Multimedia
AUTORENSYSTEM
EGO

für
Windows™

REFERENZHANDBUCH
2.11

Deutsche Bearbeitung:

Mag.Ing. Hermann LEUSTIK

Vorwort

Kommentar [NP1]: Page: 1

Diese Dokumentation enthält eine Beschreibung aller derzeit verfügbaren EGO-Befehle. Die Beschreibung dieser Befehle ist auch im Programm über die Hilfefunktion aufrufbar.

Darüber hinaus erhalten Sie einen Überblick über die Struktur der Module, deren Funktionsweise und einen Überblick über die Werkzeuge, welche vom Autorensystem angeboten werden.

Sollten Sie zum ersten Mal mit EGO arbeiten, dann studieren Sie zuerst das *Benutzerhandbuch*.

Navigation im Hilfesystem

Zugang und Ausstieg aus der Hilfe

Durch Betätigung der Funktionstaste F1 im Autorensystem kommen Sie in das Hilfesystem. Sie können die Hilfe auch aus dem Menü des Autorensystems aufrufen oder die entsprechende Ikone für die Hilfe anklicken.

Um eine Hilfe zu einem bestimmten EGO-Befehl aus dem Autorensystem zu erhalten, plazieren Sie den Cursor auf diesen Befehl und klicken danach auf F1.

Um die Hilfe zu verlassen, wählen Sie die Taste Exit aus der Tastenleiste (unterhalb des Titelbalkens oder führen einen Doppelklick auf der Ikone in der linken oberen Ecke des Hilfefensters durch. Wenn Sie die Tastatur benutzen, können Sie auch ESC oder Alt-F4 drücken.

Navigation

Mit der Tastatur:

↑, ↓	Nach oben, nach unten bewegen
PgUp, PgDn	Mehrere Zeilen nach oben, unten bewegen
←, →	Vorhergehende Seite, nächste Seite

Darüber hinaus ist auch ein direkter Sprung zu einer Seite möglich. Wenn Sie eine Zahl eingeben, erscheint plötzlich eine Editorzeile, mit deren Hilfe die Seitennummer übernommen wird, die Sie eingegeben haben. Durch Drücken auf die Eingabetaste erhalten Sie die gewünschte Seite. Dabei handelt es sich um die verkürzte Form der Option Gehe zu Seite... im Menü der Hilfe (siehe weiter unten).

Mit der Maus:

Eine Steuerungsleiste befindet sich in der rechten oberen Ecke des Hilfe-Fensters.

Hit Hilfe der Tasten wählen Sie die vorhergehende oder nächste Seite.

In der Thementabelle und im Index bringt Sie ein Klick auf das gewünschte Item in den entsprechenden Teil des Dokumentes.

Wenn Sie auf die Taste Dokument klicken, kommen Sie an den Anfang des Dokumentes. Dies ist derselbe Effekt, als wenn Sie im Menü der Hilfe auf das Item An den Anfang klicken.

Hypertext-Sprung:

Hypertext-Wörter erscheinen in der Farbe blau. Ein Hypertext-Wort ist direkt mit einer anderen Seite verbunden. Wenn Sie auf ein Hypertext-Wort klicken, so kommen Sie direkt zur ersten Seite, wo dieses Wort beschrieben ist. Wenn Sie erneut auf dieses Hypertext-Wort klicken, springen Sie sequentiell zu allen Seiten, in denen dieses Wort vorkommt.

Das Menü des Hilfesystems

Rufen Sie das Menü auf, indem Sie auf die Taste klicken, oder indem Sie Ctrl-i drücken.

Dieses Menü ermöglicht:

- die direkte Anwahl einer Seite
- das Zurückkehren zur zuletzt gesehenen Seite
- den Zugang zum Histogramm der Beratungen (die letzten 32 betrachteten Seiten)
- das Hinzufügen von Fußnoten
- das Drucken einer oder mehrerer Seiten
- das Wechseln von System-Einstellungen

Einige dieser Optionen sind auf verkürztem Wege auch durch die Aktivierung von Tasten durchführbar.

Fußnoten

Wenn eine Seite des Dokumentes eine Fußnote beinhaltet, erscheint die Ikone in der rechten oberen Ecke des Hilfefensters. Jede Seite kann nur eine einzige Fußnote enthalten.

Um die Fußnote einer Seite anzuzeigen, wählen Sie die Option Fußnote im Menü des Hilfefensters oder führen einen Doppelklick auf der Ikone durch. Die Fußnote wird in einem Editierungsfenster dargestellt, wo sie gelesen, aber auch ergänzt bzw. verändert werden kann.

Um einer Seite eine Fußnote hinzuzufügen, wählen Sie im Menü zuerst die Option Fußnote und verfassen danach Ihren Text im Editierungsfenster.

Die Fußnoten sind nicht in der Hilfedatei, sondern in separaten Dateien gespeichert. Das bedeutet, wenn Sie EGO auf einer anderen Maschine installieren, müssen Sie alle Fußnoten-Dateien (mit der Dateierweiterung *.ANN) manuell in das neue EGO-Verzeichnis übertragen.

Kopieren - Anhängen

Um eine Seite oder einen Teil einer Hilfe-Seite zu kopieren und einem Modul anzuhängen, ist es erforderlich, diese Zone mit Hilfe der Maus zu markieren. Danach muß ins Editierungsfenster des Moduls gewechselt werden. Durch Drücken von Shift-Ins wird der Text an der Position des Cursors eingefügt.

Editor des Autorenmodus

Bewegen des Cursors

←, ↑, ↓, →	Nach links, nach oben, nach unten, nach rechts
PgUp, PgDn	Vorhergehende Seite, folgende Seite
↖, Ende	Zeilenanfang, Zeilenende
TAB, Shift+TAB	Mit dem Tabulator nach rechts, nach links (um 8 Positionen)
Ctrl+PgUp, Ctrl+PgDn	Horizontales verschieben nach rechts und nach links
Ctrl+↖, Ctrl+Ende	Anfang/Ende des Arbeitsblattes
Ctrl → und Ctrl ←	Wortweise nach rechts oder links

Einfache Editierung

Ins	Einfügen / Überschreiben
Del, Backspace	Löschen eines Zeichens
Ctrl+J	Verbinden von 2 Zeilen
Ctrl+L	Einfügen einer Zeile
Ctrl+S	Löschen einer Zeile oder eines Analyseblocks
Ctrl+V	Teilen einer Zeile
Alt+Backspace	Rückgängig machen / Wiederherstellen (Undo)

Textblock

Die Cursor-Tasten ← , ↑ , ↓ , →, PgUp, PgDn in Verbindung mit der SHIFT-Taste ermöglichen das Erstellen eines Text-Blocks.

Ctrl+Shift→	Auswählen eines Wortes
Ctrl+Ins	Den Text-Block in die Zwischenablage kopieren
Shift+Del	Den Text-Block in die Zwischenablage kopieren und vom Bildschirm löschen
Shift+Ins	Den Textblock aus der Zwischenablage einfügen
Ctrl+Del	Einen Textblock löschen (ohne Speicherung in der Zwischenablage)

Spezielle Optionen unter EGO

Ctrl-´.'´	Beginn eines Befehls
F1	Hilfe
F2	Wechseln zwischen "Text" und "Analyse" und umgekehrt
F3	Detaillierte Anzeige in der Analyse (Zoom)
F4	Einfügen eines Analyseblocks
F5	Ausführen eines Moduls
F6	Liste der im Moment editierten Module
F7	Einfügen / editieren einer Notiz
F8	Suchen nach einer Zeichenkette
Alt F8	Suchen und ersetzen einer Zeichenkette
Shift-F1	Einfügen des Operators "NEGATION" (**#**)
Shift-F2	Einfügen des Operators "ODER EXKLUSIV" (⊕)
Shift-F3	Einfügen des Operators "IMPLIKATION" (⇒)
Shift-F4	Einfügen des Operators "UND" (•)
Shift-F5	Einfügen des Operators "ODER" (**+**)
Shift-F6	Einfügen des Operators "VORHER" (←)
Shift-F7	Einfügen der " öffnenden Klammer ("
Shift-F8	Einfügen der " schließenden Klammer) "
Shift-F11	Einfügen des Zeichens «Leerzeichen»
Shift-F12	Einfügen des Zeichens «Zeilenschaltung» (Enter)

DAS MODUL

PROBLEMSTELLUNG

Es ist bekannt, daß ein Kurs aus einer Folge von Lektionen besteht.

Die Zusammenstellung eines Kurses kommt also der Ausarbeitung einer Vielzahl von Lektionen gleich. Diese Tatsache gilt auch für den computergestützten Unterricht.

Unter "Modul" versteht man in EGO eine Lektion oder einen Teil einer Lektion, die durch eine logische Struktur bereichert wurde. Diese logische Struktur besteht hauptsächlich aus Anweisungen, die das Verhalten des Systems in Abhängigkeit des Verhaltens des Schülers definieren.

DIE STRUKTUR EINES MODULS

Ein Modul besteht aus drei Hauptteilen (Abb.1):

1. Dem "TEXT"-Teil
2. Dem Interventionspunkt des Schülers (IST)
3. Dem "ANALYSE"-Teil (AT)

TEXT - TEIL
INTERVENTIONSPUNKT DES SCHÜLERS
ANALYSE - TEIL

Abb.1: Das Modul

DER TEXT - TEIL

Der "TEXT"-Teil besteht aus den Anweisungen, die eingesetzt werden, um anläßlich der Ausführung des Moduls eine "Bildschirmseite" zu erzeugen.

Diese Anweisungen können verschiedenster Art sein. Die dem Schüler gezeigte "Seite" kann folgendes enthalten:

- Text
- Grafiken
- Kurven
- Berechnungen
- Audio-visuellen Sequenzen
- Interventionspunkte des Schülers
- Text- oder Antwortfenster
- Animationen
- Simulationen
- eine Kombination der angeführten Möglichkeiten

DER INTERVENTIONSPUNKT DES SCHÜLERS

Das ist ein Eingabebereich für vom Schüler verfaßte Mitteilungen bzw. für das Anklicken von Bildschirmbereichen mit Hilfe der Maus. Dieser Eingabebereich wird von EGO automatisch erzeugt, wenn man von Text-Teil zum Analyseteil übergeht

Ein Interventionspunkt kann aber auch jederzeit im Textteil oder im Kommentar eines Analyseblocks eingefügt werden.

DER ANALYSE-TEIL

Der Analyseteil ist der dritte und letzte Teil des Moduls. Dieser Teil analysiert die Handlungen des Schülers (Mitteilungen oder Mausklick) und reagiert auf pädagogisch passende Weise.

UNTER-MODULE

Ein Modul kann auf den "TEXT"-Teil oder den "ANALYSE"-Teil beschränkt werden.

FUNKTIONSWEISE DER MODULE

FUNKTIONSWEISE DES TEXT - TEILS

Der "TEXT"-Teil eines Moduls besteht aus Zeilen, die maximal 1024 Zeichen enthalten können.

Jede dieser Zeilen, die von links nach rechts geschrieben werden, besteht aus einer Folge von Befehlen, die sequentiell ebenfalls von links nach rechts ausgeführt werden. Mit anderen Worten: Der n-te Befehl einer Zeile wird erst nach der Ausführung des Befehls (n-1) ausgeführt.

EGO führt zuerst die Zeile 1 aus, dann die Zeile 2 etc. bis zur letzten Befehlszeile des "TEXT"-Teils.

EGO führt also zuerst die Befehle der Zeile 1 von links nach rechts aus: Von B1 bis zum letzten Befehl, dann geht es zur Zeile 2, in der das Programm die vorhandenen Befehle B1 B2 etc. ausführt, um anschließend die nächste Zeile abzuarbeiten.

FUNKTIONSWEISE DES INTERVENTIONSPUNKTES DES SCHÜLERS

Sobald EGO die im Textteil vorhandenen Befehle ausgeführt hat und im Modul ein Analyseteil vorhanden ist, wartet es automatisch auf eine Aktion des Schülers, die entweder aus einer Mitteilung oder aus einem Mausklick bestehen kann.

FUNKTIONSWEISE DES ANALYSETEILS

TEXT - TEIL
INTERVENTIONSPUNKT
Antwortmodell 1
Pädagogische Sanktion (Kommentar)
Antwortmodell 2
Pädagogische Sanktion (Kommentar)
- - -
- - -
Antwortmodell N-1
Pädagogische Sanktion (Kommentar)
Antwortmodell N
Pädagogische Sanktion (Kommentar)

Abb. 2: Analyseteil

Der Analyseteil besteht aus einer Folge von "Analyseblöcken", die von 1 bis N numeriert sind.

Jeder Analyseblock besteht aus zwei Teilen:

- dem Antwortmodell

- der pädagogischen Sanktion

Sobald der Schüler eine Mitteilung m eingibt, wird diese Mitteilung m dem Antwortmodell des ersten Analyseblocks zugewiesen.

Zwei Fälle können eintreten:

a - das Antwortmodell "versteht" die Mitteilung m

b - das Antwortmodell "versteht" die Mitteilung m nicht.

Fall a: Die vorgesehene pädagogische Sanktion wird ausgeführt und der Interventionspunkt des Schülers wird reaktiviert

Fall b: Die vorgesehene pädagogische Sanktion wird nicht ausgeführt und m wird dem Antwortmodell des folgenden Analyseblocks zugeordnet.

Der gleiche Prozeß wiederholt sich von Analyseblock zu Analyseblock solange, bis ein Antwortmodell m "versteht". Darauf wird die in diesem Antwortmodell vorgesehene pädagogische Sanktion ausgeführt.

Wenn kein Antwortmodell die Mitteilung m versteht, führt EGO gleichzeitig drei Maßnahmen durch:

1. Es erscheint die Mitteilung:
   Ich habe nicht verstanden. Geben Sie bitte eine andere Antwort ein.

2. Es speichert die Mitteilung m automatisch in der Liste der nicht verstandenen Mitteilungen.

3. Es reaktiviert den Interventionspunkt des Schülers und wartet, bis dieser eine neue Mitteilung eingibt.

{ PRIVATE infrtf on }DIE VARIABLEN

Unter EGO existieren 2 Arten von Variablen:

- die GLOBALEN Variablen
- die LOKALEN Variablen

DIE GLOBALEN VARIABLEN

Wie der Name schon sagt, haben diese Variablen globale Gültigkeit, das heißt, diese Variablen können in jedem Modul der didaktischen Einheit verwendet und eingesetzt werden.

Der Name einer globalen Variablen besteht aus einer Folge von maximal 32 alphanumerischen Zeichen.

Beispiel für einen Namen einer globalen Variablen: **Zahl**

DIE LOKALEN VARIABLEN

Die lokalen Variablen haben nur in dem Modul, in dem sie erstellt werden Gültigkeit. Das bedeutet, sie existieren nur in dem Modul, in dem sie erstellt werden und verschwinden anläßlich einer Verbindung mit einem anderen Modul.

Der Name einer lokalen Variablen ist durch eine Folge von max. 31 alphanumerischen Zeichen definiert, denen das Zeichen "_" (unterstrichenes Leerzeichen) vorangeht.

Beispiel für den Namen einer lokalen Variablen: **_Klick**

ANMERKUNG

Zwischen Großbuchstaben und Kleinbuchstaben besteht kein Unterschied. Was eine globalen Variablen von einer lokalen unterscheidet, ist das unterstrichene Leerzeichen vor dem Namen einer lokalen Variablen.

TYPEN VON VARIABLEN

Es gibt vier Typen von globalen oder lokalen Variablen:

- Numerische Variable
- Variable vom Typ Zeichenketten
- Variable vom Typ Zahlentabellen
- Zeichenketten-Dateien

DIE NUMERISCHEN VARIABLEN

Numerische Variable sind Variable, die reelle ganze Zahlen oder Dezimalzahlen zwischen $1.7*10^{-308}$ und $1.7*10^{308}$ enthalten können.

VARIABLEN VOM TYP ZEICHENKETTE

Die Variablen vom Typ Zeichenkette (im weiteren auch als ZKette bezeichnet) sind Variable die bis zu 1024 alphanumerische Zeichen enthalten können.

VARIABLEN VOM TYP ZAHLENTABELLEN

Die Variablen vom Typ "Zahlentabellen" sind Tabellen, die numerische Variablen enthalten. Eine Tabelle kann bis zu 8000 Werte enthalten.

ZEICHENKETTEN-DATEIEN

Zeichenketten-Dateien sind Dateien, die Variablen vom Typ Zeichenkette enthalten. Jede Datei für Zeichenketten kann eine unbeschränkte Anzahl von Zeichenketten mit maximal je 1024 Zeichen enthalten.

SYSTEMVARIABLEN

Im Programm EGO sind einige numerische Variablen und einige Variablen vom Typ Zeichenkette vordefiniert, in die für EGO benötigte Werte gespeichert werden (z.B. die Koordinaten der Mausposition).

Diese Systemvariablen werden weiter unten aufgelistet.

VERWENDUNG DER VARIABLEN

Eine Variable existiert, sobald ihr ein numerischer oder alphanumerischer Wert zugeordnet wird. Dies gilt nicht für die Variablen vom Typ "Zahlentabellen", die vor der Zuordnung erst dimensioniert werden müssen.

Eine Variable nimmt den Typ des Wertes an, der ihr beim ersten Mal zugeordnet wurde. Dies gilt nicht für die Variablen vom Typ "Zahlentabellen", die immer numerische Variable sind. Zeichenkettendateien sind immer alphanumerisch.

Um z.B. eine numerische Variable mit dem Namen Zahl mit dem Wert 0 zu erzeugen, schreibt man
: **[Zahl:=0]**

oder um eine Variable vom Typ "Zeichenkette" mit dem Namen Text zu erstellen:
[Text:=´Dies ist eine Zeichenkette´]

Anzumerken ist, daß die beiden Zeichen ":=" eine Zuordnungsbeziehung darstellen und der Ausdruck zwischen eckigen Klammern stehen muß.
Im Falle einer Variablen des Typs "Zeichenkette" muß der alphanumerische Ausdruck zwischen Apostrophe geschrieben werden. Wenn der alphanumerische Ausdruck selbst einen Apostroph enthält, so ist dieser zu verdoppeln.

In den Variablen vom Typ "Zahlentabellen" kann man ein Element durch seine Zeilen- und Spaltenindizes ansprechen.
Gegeben sei eine Tabelle Tab mit 10 Zeilen und 10 Spalten. Um den Wert 25 dem Element mit dem Zeilenindex 2 und dem Spaltenindex 3 zuzuordnen, muß man schreiben:
[Tab{2,3}:=25]

Anzumerken ist, daß gemäß der Syntax die Indizes zwischen geschwungenen Klammern stehen müssen ({Z,S}).

Für die Zeichenkettendatei kann eine Zeichenkette durch ihre Nummer angesprochen werden.
Gegeben sei eine Zeichenkettendatei mit dem Namen Adresse. Um der dritten Zeichenkette einen Inhalt zuzuordnen, muß man schreiben:
[Adresse{3}:=´Leustik, Schwanein, Klagenfurt, Austria´]

Anzumerken ist, daß der Syntax entsprechend der Index zwischen geschwungenen Klammern stehen muß ({N}).
Zu beachten ist, daß der Name einer Zeichenkettendatei aus nicht mehr als 8 Zeichen bestehen darf.

Anmerkung:

Es ist möglich, Zeilen einer Text-Datei (ASCII) zu lesen und anzuzeigen oder diese Zeichenketten zuzuweisen. Die Syntax ist dieselbe wie für das Lesen in einer Zeichenkettendatei, außer daß die Dateierweiterung einer Textdatei hinzufügt werden muß.
Beispiel:
[Liesmich.txt{2}]

oder auch
[Zeile:=Liesmich.txt{4}]

Achtung: Es ist nicht möglich in eine Text-Datei zu schreiben.

SYSTEMVARIABLE

ewCumul	Zeichenkette, die die Anhäufung der Übersetzungen enthält
ewMessage	Zeichenkette, die die Mitteilung des Schülers enthält
ewMessEdit	Zeichenkette, die die editierte Mitteilung des Schülers enthält
ewMessAnalyse	Zeichenkette, die die für die Analyse umformatierte Mitteilung des Schülers enthält
ewComplement	Zeichenkette, die die nicht übersetzten Worte der Mitteilung des Schülers enthält
ewCar	Zeichenkette, die den eben eingegebenen Buchstaben des Schülers enthält
ewTouche	Beinhaltet den Tastatur-Code der Taste, die gedrückt wurde
ewClavier	Beinhaltet den Zustand der Tastatur (•**Tastatur()**)
ewScancode	Beinhaltet den geographischen Code der gedrückten Taste
ewControle	Zeichenkette, die den Namen der Taste enthält, die gedrückt wurde oder der Zone, die angeklickt wurde
ewZone	Beinhaltet die Koordinaten der Zone, in der sich der Cursor befindet
ewIncompris	Meldung: "Ich habe nicht verstanden. Geben Sie bitte eine andere Antwort ein."
ewRamasse	Zeichenkette, die die Variable des Einsammlers enthält
ewListeMotsMult	Zeichenkette, die die Liste der Mehrfachworte enthält
ewChemin	Zeichenkette beinhaltet den Pfad des in Ausführung befindlichen Moduls
ewCarMax	Maximale Anzahl an Zeichen, die im Antwortfeld eingegeben werden können
ewClicX	Koordinate X des Mausklicks
ewClicY	Koordinate Y des Mausklicks
ewSourisX	Koordinate X der Maus
ewSourisY	Koordinate Y der Maus
ewSourisB	Nummer der Maustaste
ewDblClicX	Koordinate X des Mausdoppelklicks
ewDblClicY	Koordinate Y des Mausdoppelklicks
ewScur	Nummer des aktiven Mauscursors (•**MausCurs()**)
ewSlimite	Beinhaltet die Koordinaten der Zone, innerhalb der sich die Maus bewegen kann(•**MausZone()**)
ewSbtn	Beinhaltet die Art des Mausklicks (•**MausKlick()**)
ewMaxRamasseur	Maximale Anzahl der einzusammelnden Zahlen
ewNbNombres	Anzahl der eingesammelten Zahlen
ewRamListNbElements	Anzahl der Elemente in der Liste
ewnbMots	Anzahl der Worte in der Mitteilung des Schülers
ewnbReconnus	Anzahl der erkannten Worte
ewEvtLigne	Ausgangszeile für die Rückkehr
ewTempsMax	Antwortzeit die dem Schüler zur Verfügung steht
ewTempsPasse	Zeit die der Schüler zum Antworten benötigt hat
ewAxesActifs	Beinhaltet 1 wenn die Achsen aktiv sind, 0 wenn dies nicht der Fall ist
ewFX	Koordinate X der oberen Ecke des Antwortfensters
ewFY	Koordinate Y der oberen Ecke des Antwortfensters
ewFL	Breite des Antwortfensters.
ewFH	Höhe des Antwortfensters
ewFCT	Farbe des Textes im Antwortfenster
ewFCF	Farbe des Hintergrundes im Antwortfenster

SYSTEMVARIABLE (Fortsetzung)

ewFenRepVar	Beinhaltet den Namen der Zeichenkette bei der Editierung des Antwortfensters
ewCarMotdePasse	Beinhaltet das dargestellten Zeichen anläßlich des Aufrufens eines Passwortes
ewFCX	Koordinate X der oberen Ecke des Kommentarfensters
ewFCY	Koordinate Y der oberen Ecke des Kommentarfensters
ewFCL	Breite des Kommentarfensters
ewFCH	Höhe des Kommentarfensters
ewFCCT	Farbe des Textes im Kommentarfenster
ewFCCF	Farbe des Hintergrundes im Kommentarfenster
ewFen	Nummer des aktiven Windows-Fensters
ewFenSelect	Nummer des Windows-Fensters, das durch einen Mausklick ausgewählt wurde
ewSourisFen	Nummer des Windows-Fensters, auf dem sich die Maus befindet
ewEvtFenFermee	Liste der Fenster, deren Schließung durch ein Ereignis ausgelöst wurde
ewErreur	Nummer des Fehlers bei der Ausführung
ewErreurInfo	Beinhaltet den Parameter, der den Fehler bei der Ausführung ausgelöst hat
ewAxesActifs	Beinhaltet 1, wenn Achsen aktiv sind, sonst 0 (•**Achsen()**)
ewDefAxes	Beinhaltet die Parameter der definierten Achsen (•**AchsenDef()**)
ewHyperTexte	Angeklicktes Hypertext-Wort
ewHyperValeur	Beinhaltet den Wert der am Hypertext-Wort durchgeführten Aktion
ewHyperControle	Beinhaltet den fünften Parameter des Befehls •Hyp()
ewHyperZone	Beinhaltet die Koordinaten der Zone, welche das angeklickte Hypertext-Wort besetzt
ewHypStyle	Beinhaltet die parameter des Hypertext-Stils (•**HypStil()**)
ewPolice	Beinhaltet den Namen des aktiven Fonts
ewPolBtn	Beinhaltet den Namen des Fonts der aktiven Taste
ewPolRep	Beinhaltet den Namen des aktiven Fonts des Antwortfensters
ewDiacrit	Beinhaltet den Zustand der Sonderzeichenoptionen (•**Sonderzeichen()**)
ewMajusc	Beinhaltet den Zustand der Optionen Großbuchstaben (•**GrBuchst()**)
ewMinusc	Beinhaltet den Zustand der Option Kleinbuchstaben (•**KlBuchst()**)
ewPonct	Beinhaltet den zustand der Option Zeichensetzung (•**ZeiSetzung()**)
ewPosRep	Beinhaltet die Position des Cursors im Antwortfenster
ewPosEdit	Beinhaltet die Position des Cursors im Antwortfenster (•**TasteEdit()**)
ewWParam	Beinhaltet den ersten Parameter der Windows-Meldung
ewLParam	Beinhaltet den zweiten Parameter der Windows-Meldung
ewProcessCourant	Beinhaltet die Nummer des laufenden Prozesses
ewVariable	Beinhaltet den Variablennamen, der die Aktion ausgelöst hat (•**AktionVar()**)
ewTxtMode	Beinhaltet den Zustand der Option des freien Text-Modus (•**TextMod()**)
ewPrnX	Beinhaltet die horizontale Auflösung des Druckers
ewPrnY	Beinhaltet die vertikale Auflösung des Druckers

DER RECHNER

Alle mathematischen Ausdrücke, Bedingungen, Parameterdarstellungen, Variablen-manipulationen werden von EGO vom integrierten Rechner verwaltet und müssen in eckigen Klammern geschrieben werden ([ ]).

Überall wo EGO eckige Klammern antrifft, schickt das Programm deren Inhalt zum Rechner, der diesen analysiert.

Drei Fälle können auftreten:

- Handelt es sich um eine Variablenzuweisung, so realisiert das System die Zuweisung und löscht den Klammerausdruck aus der Zeile.

  Beispiel einer Zuweisung: **[Zahl:=10, Bruch:=Zahl/2]**

- Handelt es sich um die Anzeige einer Variablen oder eines Rechenergebnisses, so wertet das System die Anzeige aus und ersetzt den Klammerausdruck durch das Resultat.

  Beispiele einer Anzeige: **[Bruch +(Zahl*5)/2]**

- Handelt es sich um eine Bedingung, so wertet das System den Ausdruck aus und führt den Rest der Zeile aus, wenn die Bedingung erfüllt ist.

  Beispiel für eine Bedingung: **•Wenn([(A>10 ¦ Zahl=9) & (Resultat="Gut")])**

ARITHMETISCHE UND LOGISCHE OPERATOREN

&	logisches UND
¦	logisches ODER
#	Negation
>	größer
>=	größer oder gleich
<	kleiner
<=	kleiner oder gleich
<>	ungleich
:=	Zuweisung
+	Addition
-	Subtraktion
*****	Multiplikation
/	Division
^	Exponent (Potenz)
//	Modulo
!	Faktorielle

WINKELFUNKTIONEN

asin(a)	Arcsinus
acos(a)	Arccosinus
atan(a)	Arctangens
atan2(y,x)	Arctangens von y/x
cos(a)	Cosinus
cosh(a)	Cosinus hyperbolikus
exp(a)	Exponentielle
ln(a)	natürlicher Logarithmus
log(a)	dezimaler Logarithmus
sin(a)	Sinus
sinh(a)	Sinus hyperbolikus
sqrt(a)	Quadratwurzel
tan(a)	Tangens
tanh(a)	Tangens hyperbolikus

FUNKTIONEN DES RECHNERS

abs(N)	Absolutbetrag von N
and(A,B)	Diese Funktion gibt das Resultat von binärem UND zwischen A und B wieder
asc('C')	Diese Funktion gibt den Wert des ASCII-Zeichens 'C' wieder
ascha(N)	Diese Funktion gibt das Zeichen wieder, dessen ASCII-Wert N ist
borne(N,b,h)	Diese Funktion limitiert eine Zahl N in einem Bereich zwischen den Endpunkten b und h Wenn N<b, dann N=b Wenn N>h, dann N=h Wenn b<=N<=h, dann N=N
cadre(N,b,h)	Liefert das Ergebnis "RICHTIG", wenn N ein Wert zwischen b und h (inklusive h) ist. (RICHTIG=1, FALSCH=0)
car(N)	Eigenschaft von N, liefert: -1, wenn N<0 0, wenn N=0 1, wenn N>0
chalong(ZKetteName,´Trennz´)	Liefert die Länge (Anzahl der Worte) der Zeichenketten-Liste, wenn ein Trennzeichen ´Trennz´ im Befehl nicht vorkommt, die Länge (Anzahl der Zeichen) der Zeichenkette. Das Trennzeichen, falls vorhanden, muß immer zwischen Apostrophen stehen.
chamot(ZKetteName,N,´Trennz´)	Liefert das N-te Wort oder das N-te Zeichen einer Zeichenkette. Das Trennzeichen, fall vorhanden, muß immer zwischen Apostrophen stehen.
chaind(ZKetteName,Wort,´Trennz´)	Liefert den Index des Wortes oder Zeichens einer Zeichenkette. Das Trennzeichen, falls vorhanden, muß immer zwischen Apostrophen stehen
chaextract(ZKetteName,N,Nb,´Trennz´)	Liefert die Anzahl Nb der Worte oder Zeichen ab der Position N der Zeichenkette. Das Trennzeichen, falls vorhanden, muß immer zwischen Apostrophen stehen

FUNKTIONEN DES RECHNERS (Fortsetzung)

cor(NameTab)	Liefert den linearen Korrelationskoeffizienten der Elemente zweier Zeilen einer Tabelle in zwei Dimensionen
dir(N)	Liefert: 0, wenn N<=0 1, wenn N>0
fexist(´Dateiname´)	Diese Funktion erlaubt, das Vorhandensein einer Datei zu überprüfen. Sie liefert "RICHTIG", wenn die Datei "Dateiname" existiert und "FALSCH", wenn sie nicht existiert. Wenn der Dateiname in Buchstaben vorhanden ist, dann muß er zwischen Apostrophen stehen. Beispiel: **[egoinst:=fexist('c:\windows\ego.ini')]** Wenn der Dateiname in einer Zeichenkette vorhanden ist, dann kann er direkt ohne Apostrophe und Klammern eingeben werden. Beispiel: **[DName:='c:\egowin\ego.ini',inst:=fexist(DName)]**
fnbcha(Dateiname)	Diese Funktion ermittelt die Anzahl der Zeichenketten in einer Datei
flncha(Dateiname)	Diese Funktion ermittelt die Größe der Zeichenketten in einer Datei. Genauer gesagt die maximale Anzahl von Zeichen, die eine Zeichenkette in dieser Datei enthalten kann
has(Zu,Zo)	Zieht eine Zufallszahl zwischen Zu und Zo (exklusive Zo)
hasf(Zu,Zo)	Zieht eine Dezimalzahl zwischen Zu und Zo (exklusive Zo)
hex('H')	Diese Funktion gibt den Dezimalwert einer Zeichenkette 'H' wieder, welche den hexadezimalen Ausdruck einer Zahl beinhaltet
hexcha(N)	Diese Funktion gibt eine Zeichenkette wieder, welche den hexadezimalen Ausdruck der Zahl N beinhaltet
max(A,B)	Ermittelt das Maximum von A und von B
min(A,B)	Ermittelt das Minimum von A und von B

FUNKTIONEN DES RECHNERS (Fortsetzung)

mog(Tabellenname)	Ermittelt das geometrische Mittel der in der Tabelle "Tabellenname" enthaltenen Elemente
moh(Tabellenname)	Ermittelt das harmonische Mittel der in der Tabelle "Tabellenname" enthaltenen Elemente
num(Variable)	In einer Zeichenkette plaziert, bewirkt dieser Befehl, daß der Inhalt der numerischen Variablen an der Stelle eingesetzt wird
not(N)	Dieser Befehl gibt die binäre Negation der Zahl N wieder
or(A, B)	Dieser Befehl gibt das Resultat von binär ODER zwischen A und B wieder
pair(N)	Liefert 1, wenn N eine gerade Zahl ist, 0 wenn nicht
rnd(N)	Rundet die Zahl N
rla(Tabname), rlb(Tabname)	Ermittelt die Werte von a und b, Parameter von y=ax+b, lineare Regressionsgerade, die durch eine Punktwolke gelegt wird, deren x-Werte eine Zeile der Tabelle und deren y-Werte eine andere Zeile der Tabelle mit 2 Dimensionen belegen
sig(Tabellenname)	Liefert die Standardabweichung der Elemente der Tabelle "Tabellenname"
sqrt(N)	Quadratwurzel von N
sys(N,FenNum)	Ermittelt folgende Systemwerte N (benötigt bei DLL Funktionen) N = 0 liefert das "Handle" des Fensters FenNum N = 1 liefert die horizontale Auflösung N = 2 liefert die vertikale Auflösung N = 3 liefert die Anzahl der Farben N = 4 liefert die aktuellen Einstellungen des Fensters FenNum N = 5 liefert das DC des Fensters FenNum N = 6 liefert das «Handle» der aktiven Farbpalette N = 7 liefert die horizontale Auflösung des Moduls N = 8 liefert die vertikale Auflösung des Moduls N = 9 liefert die Windows-Version N = 10 liefert die EGO-Version

FUNKTIONEN DES RECHNERS (Fortsetzung)

tabsl(TabName,ZNummer,Tiefe)	Ermitteln der Summe der Elemente der Zeile "ZNummer" (und der Ebene "Tiefe" im Falle einer dreidimensionalen Tabelle) der Tabelle "TabName"
tabsc(TabName,SpNummer,Tiefe)	Ermitteln der Summe der Elemente der Spalte "SpNummer" (und der Ebene "Tiefe" im Falle einer dreidimensionalen Tabelle) der Tabelle "TabName"
tba(N)	Abrunden von N auf die nächste ganze Zahl
tha(N)	Aufrunden von N auf die nächste ganze Zahl
UnitType('C')	Diese Funktion liefert den Laufwerkstyp des eingebundenen Speichers, angegeben durch den in Anführungzeichen gesetzten Buchstaben . Die erhaltenen Werte können Werte zwischen 0 und 6 einnehmen: 0 = Kein Laufwerk mit diesem Laufwerksnamen 1 = Wechsellaufwerk 2 = Festplatte fix 3 = entferntes Laufwerk (Netzwerk) 4 = CD-ROM 5 = Diskettenlaufwerk 6 = RAM-Disk
xor (a,B)	Diese Funktion liefert das Resultat von ODER Exclusiv binär zwischen A und B

DIE BEFEHLE

EGO bietet eine Reihe leistungsfähiger Befehle an, die einfach zu verwenden sind:

Ein Befehl wird durch seinen Namen definiert, an dessen Beginn ein bestimmtes Zeichen (•) stehen muß, das man durch die Tastenkombination **[Ctrl] + [.]** erhält; danach stehen in Klammern die Parameter.

Sollte ein Befehl keine Parameter enthalten, so ist der Befehl mit einer öffnenden und schließenden Klammer zu beenden (z.B.: •**Eingabe**()).

DIE FUNKTIONEN IM NETZWERK

1) Die notwendige Umgebung

Die Netzwerkfunktionen von EGO Windows basieren auf dem Protokoll IPX von Novell. Also können die Verwaltungen der Netzwerke Lite, Personal Novell, Novell Server 3.1x und Novell Server 4.1x die Netzwerkfunktionen von EGO Windows nutzen.

2) Die Philosophie

Zwei Konzepte sind bei EGO Windows in Gebrauch:

a) Der Kanal

Ein Kanal korrespondiert mit einer Frequenz, mit der man EGO Programme senden und empfangen kann. Um EGO Zeichenketten auf einem Kanal senden und empfangen zu können, ist es notwendig, sich an diesen Kanal anzuschließen.

b) Der Name

Anläßlich der Verbindung mit einem Kanal muß ein Anwendername angeben werden, wobei dieser Name auf diesem Kanal nur einmal vorkommen kann. Man kann den Namen des Schülers verwenden oder eine andere Identität. EGOWin überprüft, ob dieser Name auf diesem Kanal nicht bereits verwendet wird.

Anmerkung:
- Man kann sich auf einem Kanal unter Verwendung unterschiedlicher Namen mehrmals einloggen.
- Man kann sich auf mehreren Kanälen mit dem selben Namen oder mit unterschiedlichen Namen einloggen.
- Man kann bis zu 128 Verbindungen aufbauen.

Zu jedem Zeitpunkt kann man unter EGO die Liste der verbundenen Anwender abfragen und wenn man verbunden ist, EGO-Programme senden und empfangen.

Das Übertragen von Zeichenketten erfolgt auf Grund von EGO-Befehlen, die man mit einem Kanal, mit einer Identifikation des Absenders und des Empfängers und der Zeichenkette versieht. Die Nummer des Kanals muß angegeben werden, da man seine Verbindungen auf mehreren Kanälen aufbauen kann. Die Identifikation des Absenders muß angegeben werden, da auf einem Kanal mehrere Verbindungen aufgebaut werden können.

ÜBERSICHT ÜBER DIE BEFEHLE

Die folgende Liste enthält die EGO-Befehle nach Kategorien zusammengefaßt:

ALLGEMEINE BEFEHLE:

- **Aktion**() Verwaltung von Aktionen
- **AktionMeld**() Aktionsverwaltung für Windows-Meldungen
- **AktionVar**() Aktionsverwaltung bei Verwendung von Variablen
- **AktivExtern**() Aktiviert eine Anwendung
- **AnalyseFort**() Fortsetzen in der Analyse
- **AntwFen**() Definieren eines Antwortfensters
- **AntwFenBef**() Definieren von Listen speziellerTasten für ein Antwortfenster
- **AntwFenOpt**() Definieren der Optionen für das Verlassen eines Antwortfensters
- **Aufgabe**() Starten einer Aufgabe
- **AufgEnde**() Beenden der Ausführung einer Aufgabe
- **AufgFort**() Fortsetzen einer vorübergehend angehaltenen Aufgabe
- **AufgPause**() Vorübergehendes Anhalten einer Aufgabe
- **BedAM**() Bedingung in einem Antwortmodell
- **BildDim**() Ermitteln der Abmessungen eines Bildes
- **BschDruck**() Drucken eines Bildschirmbereiches
- **BschDruckW**() Benutzung der Windows-Zwischenablage ermöglichen / unterbinden
- **BschKopie**() Kopie zwischen virtuellem und physischem Bildschirm
- **BschKopieBH**() Kopie zwischen virtuellem und physischem Bildschirm
- **Capt**() Speichern eines Bildschirmbereiches
- **CaptBH**() Speichern eines Bildschirmbereiches
- **CoAufgabe**() Aufrufen einer Co-Aufgabe
- **CoMod**() Aufrufen eines Co-Moduls
- **DateiKopie**() Kopieren einer Datei
- **DateiLösch**() Löschen einer Datei
- **DateiÖffnen**() Aktivieren eines Dialogfensters für das Öffnen von Dateien
- **DateiSichern**() Aktivieren eines Dialogfensters für das Sichern von Dateien
- **DateiUmben**() Umbenennen einer Datei
- **Datum**() Ermitteln des Systemdatums
- **DDEBef**() Senden von DDE-Befehlen
- **DDENachr**() Beenden des Ruhezustandes von EGO durch einen DDE-Server
- **DDESetze**() Setzen eines Wertes in einem DDE-Server
- **DDEWert**() Übernehmen eines Wertes von einem DDE-Server
- **Dir**() Auflisten des Inhalts eines Verzeichnisses
- **DLL**() Aufrufen einer DLL-Funktion
- **DoppelWorte**() Doppelworte in der Antwort des Schülers
- **Druck**() Definiert und erstellt eine Seite zum Ausdrucken
- **Eingabe**() Eine Eingabe des Schülers ermöglichen (Interventionspunkt)

ALLGEMEINE BEFEHLE (Fortsetzung)

•**EingabeErl**()	Verwaltung des Interventionspunktes des Schülers
•**Ende**()	Beenden der Ausführung eines Moduls oder eines Kurses
•**Extern**()	Ausführen eines externen Programms
•**ExternEnde**()	Beenden eines externen Programms
•**FenKopie**()	Kopieren zwischen Fenstern
•**GCursPos**()	Ermitteln der Position des Grafikcursors
•**Hyp**()	Setzen eines Hypertext-Wortes
•**HypLösch**()	Löschen von Hypertext-Wörtern
•**HypStil**()	Definieren des Stils für Hypertext-Worte
•**IPS**()	Umwandeln eines in Klammern stehenden Antwortmodells in die Inverse Polnische Schreibweise
•**KomFen**()	Definieren eines Kommentarfensters
•**ModListe**()	Liste der Module eines EGO-Prozesses
•**ModStat**()	Ermitteln des Zustandes eines Moduls oder Co-Moduls, einer Aufgabe oder Co-Augabe
•**ModName**()	Übernimmen des Namens des gerade in Ausführung befindlichen Moduls
•**Pause**()	Pause in der Ausführung eines Moduls
•**Reset**()	Rücksetzen von verschiedenen Optionen auf ihren Ursprungswert
•**Sam**()	Festlegen des Zahleneinsammlers
•**Schleife**()	Definieren einer Schleife
•**ScrollText**()	Scrollen eines Textes in einem Textfenster horizontal oder vertikal
•**Sprung**()	Springen zu einer Zeile
•**SprungBlock**()	Springen zu einem Analyseblock
•**SucheZkDatei**()	Suche in Zeichenketten-Dateien
•**System**()	Verwaltung von Verzeichnissen (Directory)
•**TabCha**()	Kopieren einer Zahlentabelle in eine Zeichenkettendatei
•**TabDim**()	Ermitteln der Dimensionen einer Zahlentabelle
•**TabFüllen**()	Füllen einer Tabelle
•**TabKopie**()	Kopieren einer Tabelle in eine andere Tabelle
•**TabOrd**()	Ordnen der Werte einer Tabelle
•**TabZufall**()	Füllen einer Tabelle mit Zufallszahlen
•**TCursPos**()	Ermitteln der Position des Textcursors
•**Ton**()	Ausgeben eines Signal-Tones
•**Uhrzeit**()	Ermitteln der Systemzeit (Uhrzeit)
•**UhrzeitSyst**()	Stellt die System-Uhrzeit wieder her
•**ÜbersEinf**()	Einfügen einer Übersetzung in die Anhäufung der Übersetzungen
•**ÜbersLösch**()	Löschen von Übersetzungen in der Anhäufung der Übersetzungen
•**VarCha**()	Speichern aller globalen Variablen in eine Zeichenkettendatei
•**Weiter**()	Verbinden mit dem nächsten Modul
•**Wenn**()	Bedingung im Text- oder Kommentarteil
•**Wörterbuch**()	Aufrufen einer Wörterbuchfunktion
•**Xfi**()	Vergleichen eines Antwortmodells mit einer Reihe von Zeichenketten

BEFEHLE FÜR DIE BILDSCHIRMDARSTELLUNG

•**Achsen**()	Aktivieren bzw. desaktivieren von Achsen
•**AchsenDef**()	Definieren von Achsen
•**AltTab**()	Erlauben/unterbinden der Verwendung von Alt+Tab und Ctrl+Esc
•**AntwFenFont**()	Laden eines Fonts (Schrift) für das Antwortfenster
•**AntwPos**()	Positioniert den Cursor in einem Antwortfenster
•**Ausrichten**()	Ausrichtung des Textes
•**BHDiv**()	Definieren der Divisoren für die Anzeige von verschieden Bildschirmwippen
•**BildEntf**()	Entfernen eines Bildes aus dem Arbeitsspeicher
•**BildEntfA**()	Entfernen aller Bilder aus dem Arbeitsspeicher, außer den angeführten
•**BildMax**()	Festlegen der maximalen Anzahl an geladenen Bildern
•**BildName**()	Ermitteln des Namens eines im Arbeitsspeicher geladenen Bildes
•**BildZahl**()	Ermitteln der Anzahl an geladenen Bildern
•**Bsch**()	Auswählen des aktiven Bildschirmes
•**BschListe**()	Liefert die Liste der existierenden virtuellen Bildschirme
•**BschLösch**()	Löschen des Bildschirms
•**Clip**()	Aktivieren bzw. desaktivieren eines Bildschirmbereiches
•**ClipDef**()	Definieren eines Bildschirmausschnittes (Clipping)
•**DLLFrei**()	Schließen einer DLL (Dynamic-Link Library)
•**DLLLaden**()	Laden einer DLL
•**FarbeFrei**()	Definieren einer verfügbaren Farbskala für Animationen
•**FarbenRes**()	Festlegen der Farben im reservierten Modus
•**Font**()	Laden eines Fonts (Schrift) für den Text
•**FontInst**()	Installieren einer TRUE TYPE Schrift
•**FontLösch**()	Deinstallieren eines TRUE TYPE Schrift
•**FontVorh**()	Überprüft, ob ein bestimmter TRUE TYPE Font installiert ist
•**Füllfarbe**()	Definieren der Füllfarbe
•**Grafikformat**()	Definieren des aktuellen Grafikformates
•**Komma**()	Festlegen der Anzeigegenauigkeit von Zahlen
•**MausCurs**()	Definieren des Mauscursor-Typ
•**MausCursB**()	Festlegen eines persönlichen Mauscursors
•**MausKlick**()	Simulieren eines Maus-Klicks
•**MausPos**()	Positionieren des Mauscursors
•**MausZone**()	Definieren der Bildschirmzone, innerhalb der die Maus bewegt werden kann
•**ModParam**()	Wiedererlangen der Parameter eines vorherigen Moduls
•**ModVerb**()	Erzwungene Verbindung mit einem Modul
•**NichtVerst**()	Setzen des Pfades der Datei mit den nicht verstandenen Mitteilungen
•**Nullen**()	Anhängen von Nullen am Ende von Zahlenwerten
•**Pinsel**()	Definieren der Pinselfarbe
•**PinselStil**()	Definieren des Pinselstils
•**Pos**()	Positionieren des Textes
•**Rechner**()	Freigeben bzw. sperren des Rechners
•**SamList**()	Erzeugen einer Liste zum Zahleneinsammeln

BEFEHLE FÜR DIE BILDSCHIRMDARSTELLUNG (Fortsetzung)

•**ScrollDef**()	Definieren des Scrollens in einem Kommentarfenster
•**TabDef**()	Definieren und dimensionieren einer Zahlentabelle
•**Tastatur**()	Aktiviert / desaktiviert die Tastatur oder simuliert einen Anschlag auf ein Zeichen
•**TasteFont**()	Definieren des Fonts für die Windows-Tasten
•**TasteInfo**()	Liefert die Eigenschaften einer Taste
•**TasteLösch**()	Löschen von Tasten, Menüs, Auswahltasten und Zonen
•**TasteLöschA**()	Löschen aller Tasten außer jenen, die angeführt sind
•**TasteModus**()	Definieren des Tastenmodus
•**TasteName**()	Ermitteln des Namens einer aktiven Taste oder aktiven Zone
•**TastePos**()	Positionieren des Cursors bei TasteListe und TasteScroll
•**TasteZahl**()	Ermitteln der Anzahl der verwendeten Tasten und Zonen
•**TextMod**()	Aktiviert/deaktviert den freien Textmodus
•**VBschDef**()	Definieren eines virtuellen Bildschirmes
•**VBschLösch**()	Löschen eines virtuellen Bildschirmes
•**VarLösch**()	Löschen von Variablen
•**VarName**()	Ermitteln des Namens einer verwendeten Variablen
•**VarZahl**()	Ermitteln der Anzahl der verwendeten Variablen
•**ZeileNum**()	Übernimmt die Nummer der Zeile in der dieser Befehl ausgeführt wird
•**ZeiMod**()	Setzen des Modus für das Zeichnen

BEFEHLE FÜR ZEICHENKETTEN

•**ChaDatei**()	Definieren einer Zeichenkettendatei
•**ChaDateiEinf**()	Fügt eine Zeichenkette in einer Zeichenketten-Datei ein
•**ChaDateiLösch**()	Löscht eine Zeichenkette in einer Zeichenketten-Datei
•**ChaErsetzen**()	Ersetzen jedes Vorkommens eines Wortes in einer Liste
•**ChaExclus**()	Heraussuchen derjenigen Elemente aus der Liste C1, die nicht in der Liste C2 enthalten sind
•**ChaIdent1**()	Suchen der gemeinsamen Elemente aus den Listen C1 und C2, die auch an gleicher Position stehen
•**ChaIdent2**()	Suchen der gemeinsamen Elemente aus den Listen C1 und C2
•**ChaLösch**()	Löschen eines Wortes in einer Variablen Liste
•**ChaOrd**()	Ordnen einer Zeichenkette oder Zeichenkettendatei
•**ChaOrdCha**()	Ordnen einer Liste durch eine weitere Liste
•**ChaOrdTab**()	Ordnen einer Liste durch eine Tabelle
•**ChaPos**()	Schreiben eines Wortes an einer angegebenen Position in einer Variablen Liste
•**ChaSam**()	Einsammeln von Zahlen einer Zeichenkette
•**ChaTab**()	Kopieren einer Zeichenkettendatei in eine Zahlentabelle
•**ChaTxt**()	Kopieren einer Zeichenkette in eine Textdatei
•**ChaVork**()	Ermitteln der Anzahl des Vorkommens eines Wortes in einer Liste

BEFEHLE ZUR EDITIERUNG EINER MITTEILUNG

•**GrBuchst**()	Umwandeln in Großbuchstaben
•**KlBuchst**()	Umwandeln in Kleinbuchstaben
•**Sonderzei**()	Umwandeln von Sonderzeichen in äquivalente Zeichen
•**ZeiSetzung**()	Unterdrücken der Zeichensetzung (Interpunktion)

MULTIMEDIA-BEFEHLE

•**AAPLAY**()	Spielen einer Autodesk Animator Animation (*.FLC)
•**AASTOP**()	Beenden einer Animation, die mit AAPLAY gestartet wurde
•**FLCEnde**()	Beenden einer FLC-Animation
•**FLCInfo**()	Kleines Hilfe-Fenster, gebunden an eine Taste
•**FLCLaden**()	Laden einer FLC-Animation
•**FLCListe**()	Ermitteln der Liste der Alias der geladenen FLC-Animation
•**FLCPause**()	Setzen einer FLC-Animation in Pause-Funktion
•**FLCSpiel**()	Starten einer FLC-Animation
•**MCI**()	Setzen von Befehlen zur Multimedia-Steuerung
•**TasteMCI**()	Multimedia-Taste

GRAFIK-BEFEHLE

•**AbleitPara**()	Zeichnen der Kurve der ersten Ableitung einer Parameterfunktion
•**Ableitung**()	Zeichnen der Kurve der ersten Ableitung einer Funktion
•**AchsenZei**()	Zeichnen der Achsen-Koordinaten
•**Bild**()	Anzeigen eines Bildes
•**BildAnPal**()	Paßt ein Bild an eine vorgegebene Palette an
•**BildBH**()	Anzeigen eines Bildes
•**Bogen**()	Zeichnen eines Bogens
•**BschTransp**()	Erstellen einer Maske eines virtuellen Bildschirms
•**Ellipse**()	Zeichnen einer Ellipse
•**EllipseVoll**()	Zeichnen einer ausgefüllten Ellipse
•**FarbePF**()	Ermitteln der Füllfarbe und Pinselfarbe
•**Farbskala**()	Erzeugen einer Farbskala
•**Fläche**()	Zeichnen einer rechteckigen Fläche
•**FlächeBH**()	Zeichnen einer rechteckigen Fläche
•**Funktion**()	Zeichnen der Kurve einer Funktion
•**FunktionPara**()	Zeichnen der Kurve einer Parameterfunktion
•**Füllen**()	Ausfüllen einer Fläche mit einer bestimmten Farbe
•**Integral**()	Zeichnen der Kurve des Integrals einer Funktion
•**Integralpara**()	Zeichnen der Kurve des Integrals einer Parameterfunktion
•**Kreis**()	Zeichnen eines Kreises
•**Linie**()	Zeichnen einer Linie
•**Nadel**()	Zeichnen einer Nadel
•**Objekt**()	Anzeige eines Bildes
•**PalKopie**()	Kopieren der aktuellen Farbpalette in eine virtuelle Farbpalette
•**PalFade**()	Kopieren einer virtuellen Farbpalette in die aktuelle Farbpalette
•**PalOp**()	Operationen in physischen und virtuellen Farbpaletten
•**Polygon**()	Zeichnen eines Polygons
•**Punkt**()	Zeichnen eines Punktes
•**PunktFarbe**()	Ermitteln der Farbe eines Bildschirmpunktes
•**Rechteck**()	Zeichnen eines Rechtecks
•**RechteckBH**()	Zeichnen eines Rechtecks
•**RegSeite**()	Auswahl einer Seite einer Register-Taste
•**RGBIndex**()	RGB-Komponenten des Index einer Farbe
•**RGBWechsel**()	Wechseln einer Farbe in der aktuellen Farbpalette
•**RRechteck**()	Zeichnen eines Rechtecks mit abgerundeten Ecken
•**RRechteckBH**()	Zeichnen eines Rechtecks mit abgerundeten Ecken
•**StandPal**()	Aktivieren/desaktivieren einer Standard-Palette
•**Taste**()	Definieren einer Windows-Taste
•**TasteAuswahl**()	Definieren einer Auswahltaste
•**TasteCombo**()	Definition eines Fensters vom Typ COMBOBOX (statisch)
•**TasteComboEdit**()	Definition eines Fensters vom Typ COMBOBOX mit Editor
•**TasteComboFix**	Definieren eines Fenster vom Typ COMBOBOX mit immer geöffnetem Editor

GRAFIKBEFEHLE (Fortsetzung)

•**TasteCursor**()	Definieren einer linearen Cursor-Taste (Bildlaufleiste)
•**TasteEdit**()	Definieren einer Editier-Fenster-Taste
•**TasteFix**	Definieren einer statischen Taste
•**TasteFocus**()	Auswählen einer Taste (in den Brennpunkt setzen)
•**TasteFortschr**()	Definition einer Taste mit einer Darstellung des Fortschritts
•**TasteGrafik**()	Definieren einer Grafiktaste
•**TasteHilfe**()	Kleines Hilfe-Fenster, gebunden an eine Taste
•**TasteInfo**()	Ermitteln der Parameter einer Taste
•**TasteListe**()	Definieren eines Fensters mit Bildlaufleiste
•**TasteMCI**()	Definieren einer Multimedia-Taste
•**TasteMenü**()	Definieren einer Menüleiste
•**TasteMenüStil**()	Definieren eines Item-Stils in einem Menü
•**TasteRadio**()	Definieren einer Windows Radio-Taste
•**TasteReg**()	Definition einer Register-Taste
•**TasteScroll**()	Definieren einer Bildlaufleiste
•**Text**()	Definieren von Grafiktext
•**Transp**()	Erzeugen einer transparenten Maske
•**Vektor**()	Zeichnen eines Vektors
•**WipeHor**()	Anzeigen eines virtuellen Bildschirms, der sich von links oder von rechts bewegt
•**Wmf**()	Darstellen eines Bildes (Vektorgrafik) vom Typ WMF
•**Zone**()	Definieren einer rechteckigen Zone

BEFEHLE FÜR WINDOWS-FENSTER

•**Fen**()	Aktivieren/Desaktivieren eines Windows-Fensters
•**FenDef**()	Definieren eines Windows-Fensters
•**FenFix**()	Verriegeln/Entriegeln eines Windows-Fensters
•**FenForm**()	Definieren einer nicht rechteckigen Kontur eines Windows-Fensters
•**FenInfo**()	Ermitteln der Objekt-Daten eines Windows-Fensters
•**FenList**()	Auflisten der Windows-Fenster
•**FenLösch**()	Löschen eines Windows-Fensters
•**FenOrd**()	Definieren der Anordnung von Windows-Fenstern
•**FenPos**()	Verändern und positionieren eines Windows-Fensters
•**KoordKonv**()	Konvertieren von relativen Koordinaten in ein Windowsfenster

NETZWERK-BEFEHLE

•**NwKanalAktiv**()	Erstellen der Liste aller aktiven Kanäle
•**NwKanalListe**()	Erstellen der Liste der Verbindungen auf einem Kanal
•**NwKarteNr**()	Ermitteln der Seriennummer der Netzwerkkarte
•**NwInit**()	Initialisieren der Netzwerkparameter
•**NwNachrEmpf**()	Empfangen einer EGO-Zeichenkette
•**NwNachrOK**()	Überprüfen, ob die EGO-Zeichenkette empfangen wurde
•**NwNachrQuell**()	Ermitteln der Namen des Senders und Empfängers der EGOZeichenkette
•**NwNachrSend**()	Senden einer EGO-Zeichenkette
•**NwNachrWart**()	Überprüfen, ob eine Zeichenkette in der Warteschlange ist
•**NwVerb**()	Verbinden eines Gerätes auf einem Kanal
•**NwVerbName**()	Ermitteln des Namens einer Verbindung
•**NwVerbEnde**()	Schließen einer lokalen Verbindung
•**NwVerbEndeA**()	Schließen aller lokalen Verbindungen
•**NwVerbEndeD**()	Schließen einer entfernten Verbindung

Alias: (AAPLAY) Befehl: AAPLAY

Spielen einer Autodesk Animator Animation (*.FLC)

SYNTAX •**AAPLAY(*AnimName,Xh,Yh,Xb,Yb*)**

BESCHREIBUNG Lädt und führt eine Animation des Typs FLC aus (die mit dem Programm Animator Pro oder 3D Studio erstellt wurde).

AnimName	:	Dateiname der Animation mit Angabe des Pfades
Xh,Yh	:	Koordinaten der oberen linken Ecke des Animationsfensters
Xb,Yb	:	Koordinaten der unteren rechten Ecke des Animationsfensters

ANMERKUNG Wenn nichts anderes angegeben, paßt sich das Fenster an die Größe der Animation an. Wird die Animation einmal gestartet, so läuft sie in der Schleife solange ab, bis diese mittels •**AASTOP**() angehalten wird.

Es ist möglich, mehrere Animationen zugleich zu starten

(Maximum = 10).

BEISPIEL

```
•AAPlay(D:\ANIM\ANIM1.FLC,200,100)
•AAPlay(D:\ANIM\ANIM2.FLC,400,100)
```

SIEHE AUCH •**AASTOP**() , •**FLCEnde()**, •**FLCLaden()**, •**FLCPause()**, •**FLCSpiel()**

Alias: (AASTOP) Befehl: AASTOP

Beenden einer Animation, die mit AAPLAY gestartet wurde

SYNTAX •**AASTOP**(***AnimName***)

BESCHREIBUNG Beendet eine Animation, die mit •**AAPLAY**() gestartet wurde.

AnimName : Dateiname der Animation, die mit •**AAPLAY**() gestartet wurde

BEISPIEL

```
•AASTOP(D:\ANIM\ANIM2.FLC)
```

SIEHE AUCH •**AAPLAY**(), •**FLCEnde**(), •**FLCLaden**(), •**FLCPause**(), •**FLCSpiel**()

Alias: (DerivPara) Befehl: AbleitPara

Zeichnen der Kurve der ersten Ableitung einer Parameterfunktion

SYNTAX •**AbleitPara(*f(t),g(t),Var,Vi,Vf,Inkrement,Tabelle,Stufen*)**

BESCHREIBUNG Zeichnet die repräsentative Kurve der ersten Ableitung einer Funktion in Parameterdarstellung.

f(t),g(t)	:	Gleichungen der Parameterfunktion
Var	:	Name der Variablen der Funktion
Vi	:	Anfangswert der Variablen der Funktion
Vf	:	Endwert der Variablen der Funktion
Inkrement	:	Inkrement (Veränderungswert) der Variablen, um vom Anfangswert zum Endwert zu kommen
Tabelle	:	Name der Tabellenvariablen, in der die Koordinaten der Kurvenpunkte gespeichert werden
Pdiv	:	Ein Wert der angibt, daß die Punkt-Koordinaten alle x-Stufen gespeichert werden

ANMERKUNG Die erste Ableitung wird unter Bezugnahme auf die Achsen gezeichnet, die mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man als vorletzten Parameter den Namen einer Tabelle angibt, so werden die Koordinaten der Punkte der Kurve darin gespeichert. Die Tabelle muß zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor aller zu speichernden Punkte. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•AbleitPara(y=sin(5*Teta),y=cos(3*Teta),Teta,0,6.3,0.05)
•TabDef(Tab1,200,2)•TabFüllen(Tab1,0)
•AbleitPara(x=sin(5*Teta),y=cos(3*Teta),0,6.3,0.05,Tab1)
```

SIEHE AUCH •**Funktion**(), •**Integral**(), •**FunktionPara**(), •**Ableitung**(), •**IntegralPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (Derivee) Befehl: Ableitung

Zeichnen der Kurve der ersten Ableitung einer Funktion

SYNTAX •**Ableitung(*Funktion,Var,Vi,Vf,Inkrement,Tabelle,Stufe*)**

BESCHREIBUNG Zeichnet die repräsentative Kurve der ersten Ableitung einer Funktion.

Funktion	:	Funktionsgleichung (y=f(x))
Variable	:	Name der Variablen der Funktion
Vi	:	Anfangswert der Variablen der Funktion
Vf	:	Endwert der Variablen der Funktion
Inkrement	:	Inkrement (Veränderungswert) der Variablen, um vom Wert Vi zu Vf zu kommen
Tabelle	:	Name der Tabellenvariablen, in der die Koordinaten der Kurvenpunkte gespeichert werden
Stufen	:	Ein Wert der angibt, daß die Punkt-Koordinaten alle x-Stufen gespeichert werden

ANMERKUNG Die erste Ableitung wird unter Bezugnahme auf die Achsen gezeichnet, die mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man als vorletzten Parameter den Namen einer Tabelle angibt, so werden die Koordinaten der Punkte der Kurve darin gespeichert. Die Tabelle muß zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor aller zu speichernden Punkte. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•Ableitung(y=sin(Teta),Teta,0,6.3,0.1)
•TabDef(Tab1,200,2) •TabFüllen(Tab1,0)
•Ableitung(y=2*x^2,x,-2,2,0.1,Tab1)
```

SIEHE AUCH •**Funktion**(), •**Integral**(), •**FunktionPara**(), •**AbleitPara**(), •**IntegralPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (Axes) Befehl: Achsen

Aktivieren bzw. desaktivieren der Achsen

SYNTAX •**Achsen(+)** oder •**Achsen(-)**

BESCHREIBUNG Dieser Befehl ermöglicht die Aktivierung oder Desaktivierung der mit dem Befehl •**AchsenDef**() definierten Achsen.

•***Achsen(+)***	:	aktiviert die Achsen
•***Achsen(-)***	:	desaktiviert die Achsen

SIEHE AUCH •**AchsenDef**()

Alias: (DefAxes) Befehl: AchsenDef

Definieren von Achsen

SYNTAX •**AchsenDef(*OrgX,OrgY,EinheitX,EinheitY*)**

BESCHREIBUNG Definieren der Achsenkoordinaten, auf die sich die Grafikzeichnungen beziehen (auf dem physischen oder virtuellen Bildschirm)

OrgX,OrgY : Koordinaten des Achsenursprungs

EinheitX,EinheitY : Einheiten der X- und Y-Achsen

ANMERKUNG Die Ausrichtung der Achsen wird durch das Vorzeichen der Einheiten bestimmt. Positive Einheiten der X- und Y-Achse richten die X-Achse nach rechts und die Y-Achse nach oben aus. Um zu einer Zeichnung mit absoluten Koordinaten zurückzukehren, muß man den Befehl •**Achsen**(-) verwenden.

Wird nichts angegeben, so wird die Zeichnung der Grafik auf die absoluten Bildschirmkoordinaten bezogen.

BEISPIEL

```
•AchsenDef(320,240,10,10)
•Objekt(Foto.PCX,1,1)
•Rechteck(0,0,2,3)
```

SIEHE AUCH •**Achsen**()

Alias: (TraceAxes) Befehl: AchsenZei

Zeichnen der Achsen-Koordinaten

SYNTAX •**AchsenZei(*Typ,Lx,Ly*)**

BESCHREIBUNG Zeichnet die Achsen-Koordinaten, die mit dem Befehl •**AchsenDef**() definiert wurden.

Typ	:	Typ der Achsenanzeige (0 bis 3)
Lx,Ly	:	Länge der Achsen (in Einheiten)
Typ	:	Darstellung:
0	:	Anzeige der positiven Achsenabschnitte ohne Gradeinteilung
1	:	Anzeige der gesamten Achsen ohne Gradeinteilung
2	:	Anzeige der positiven Achsenabschnitte mit Gradeinteilung
3	:	Anzeige der gesamten Achsen mit Gradeinteilung

ANMERKUNG Die Gradeinteilung der Einheiten auf den Achsen erscheint nur dann, wenn die Größe der Einheiten größer oder gleich 10 ist. Darüber hinaus werden auf den Achsen ganze Einheiten, halbe Einheiten und 10-tel Einheiten angezeigt, wenn die Einheiten größer oder gleich 20 plus ein Vielfaches von 10 sind.

Die Länge der Achsen (***Lx,Ly***) wird in der Anzahl von Einheiten angegeben.

Die Darstellung erfolgt mit der Farbe des aktuellen Pinsels.

BEISPIEL

```
•AchsenDef(200,400,30,20)
•Pinsel(248)
•AchsenZei(2,10,15)
```

SIEHE AUCH •**AchsenDef**()

Alias: (Evt) Befehl: Aktion

Verwaltung von Aktionen

SYNTAX •**Aktion(*Zeile*)** •**Aktion(*Zeile,Aktionstyp*)**

•**Aktion(*Zeile,Aktionstyp,CoMod*)** •**Aktion(+)** •**Aktion(-)**

BESCHREIBUNG Springt zu der Zeilennummer oder zu dem Label ***Zeile*** oder zu einem Co-Modul ***CoMod***, wenn die Aktion ***Aktionstyp*** eintritt.

Zeile	:	Zeilennummer oder Label
Aktionstyp	:	Aktionstyp von 1 bis 11
1	=	Taste gedrückt
2	=	Mausklick
3	=	anhaltender Mausklick
4	=	Doppelklick
5	=	Bewegung der Maus
6	=	Loslassen des Mausklicks
7	=	Schließen eines Windows-Fensters
8	=	Fehler bei der Ausführung
9	=	Bekanntgabe des Endes einer MCI-Sequenz
10	=	Sättigung einer •**TasteEdit()**
11	=	Ende eines FLC oder Ende eines FLC-Segments

ANMERKUNG Die Befehle und •**Aktion(Zeile,Aktionstyp)** definieren eine lokale Verwaltung der Aktion im Block, wo sie entstehen. Der Befehl bewirkt einen Sprung zu Zeile, wenn eine Taste gedrückt wird oder ein Mausklick erfolgt (1 und 2). Darüber hinaus speichert die Aktionsverwaltung in der Systemvariablen ewEvtLigne die Nummer der Zeile, die der Zeile folgt, die die Aktion aktiviert hat. Der Befehl •**Aktion(Zeile,Aktionstyp,CoMod)** definiert eine globale Aktionsverwaltung, was heißen soll, daß diese bis zu einem Desaktivierungsbefehl bestehen bleibt. Die so aktivierte Verwaltung läßt das Co-Modul ***CoMod*** ausführen. Am Ende des Co-Moduls angelangt, wird im aufrufenden Modul in der Zeile fortgesetzt, die der aufrufenden folgt. Der Befehl •**Aktion**(-) setzt die Verwaltung von Aktionen aus, die durch •**Aktion**(+) wieder aktiviert werden. Zu beachten ist, daß die Aktionsverwaltung in die angegebene Zeile springt und ein •**Aktion**(-) ausführt, sobald eine Aktion eintritt. Deshalb muß die Aktionsverwaltung wieder durch •**Aktion**(+) aktiviert werden, sobald die aufgetretene Aktion abgehandelt wurde. Darüber hinaus speichert die Aktionsverwaltung in der Systemvariablen ***ewEvtLigne*** die Nummer der Zeile, die jener folgt, in der die Aktion aufgetreten ist. Um die Verwaltung einer bestimmten Aktion zu desaktivieren, reicht es, diese Verwaltung mit der Nummer 0 als Zeilennummer zu definieren.

BEISPIEL

```
•Aktion(Klick,2)
••Springt zum Label Klick, wenn ein Mausklick erfolgt
```

Alias: (EvtMsg) Befehl: AktionMeld

Aktionsverwaltung für Windows-Meldungen

SYNTAX

•AktionMeld(*Zeile, MeldNum1,.....,MeldNumN*)

•AktionMeld(*Zeile, MeldNum1,.....,MeldNumN, CoMod*)

•Aktion(+)

•Aktion(-)

BESCHREIBUNG Springt zu der Zeilennummer oder zum Label Zeile oder zu einem Comodul CoMod, wenn eine der Meldungen MeldNum1, ... MeldNumN erhalten wird

Zeile : Zeilennummer oder Label

MeldNum : Nummer der Windows-Meldung

ANMERKUNG Der Befehl **•AktionMeld(Zeile,MeldNum1)** definiert eine lokale Aktionsverwaltung in dem Block, in dem er vorkommt. Darüber hinaus speichert die Aktionsverwaltung in der Systemvariablen ***ewEvtLigne*** die Nummer der folgenden Zeile, in der die Aktion ausgelöst wurde. In der Systemvariablen ewWParam wird der erste durch die Meldung übergebene Parameter, in der Systemvariablen ewLParam der zweite durch die Meldung übergebene Parameter gespeichert. Der Befehl **•AktionMeld(Zeile,MeldNum, CoMod)** definiert eine globale Aktionsverwaltung, d.h., daß diese von Modul zu Modul besteht, bis ein Befehl dies desaktiviert. Nach Beendigung des Co-Moduls wird in der Zeile fortgesetzt, die der Zeile folgt, welche die Aktion ausgelöst hat.

Der Befehl **•Aktion**(-) setzt die Aktionsverwaltung vorübergehend außer Kraft, was mit dem Befehl **•Aktion**(+) rückgängig gemacht werden kann.

Man muß festhalten, wenn eine Ereignis stattfindet, springt die Aktionsverwaltung auf die festgelegte Zeile oder in das festgelegte Co-Modul und führt dabei ein **•Aktion**(-) durch, was bedingt, daß bei Bedarf die Aktionsverwaltung mit dem Befehl **•Aktion**(+) wieder aktiviert werden muß. Um eine spezielle Aktionsverwaltung zu desaktivieren, reicht es, diese Verwaltung mit 0 als Zeilennummer umzudeklarieren.

BEISPIEL

```
•AktionMeld(MeldWin,3000)
••Springt zum Label MeldWin
••wenn die Meldung 3000 eintrifft
```

Alias: (EvtVar) Befehl: AktionVar

Verwaltung der Aktionen zur Verwendung von Variablen

SYNTAX

•AktionVar(*Zeile*,*VarName*)

•AktionVar(*Zeile*,*VarName*,,*CoMod*)

•AktionVar(+,*VarName*)

•AktionVar(-,*VarName*)

•AktionVar(-)

•Aktion(+) **•Aktion(-)**

BESCHREIBUNG Sprung auf die Zeilennummer oder auf das Label ***Zeile*** oder in ein CoModul ***CoMod***, wenn eine von den Variablen ***VarName*** verwendet wird.

Zeile : Nummer der Zeile oder des Labels

VarName : Name einer Variable

ANMERKUNG Der Befehl **•AktionVar(Zeile,VarName)** definiert eine Verwaltung von lokalen Aktionen im Block, wo sie erscheint. Außerdem nimmt die Verwaltung der Aktionen in die Systemvariable ***ewEvtLigne*** die Nummer der Zeile auf, die derjenigen folgt, in der sich die Aktionen ereignet hat. Die Systemvariable ***ewVariable*** beinhaltet den Namen der Variablen, die die Aktion bewirkt hat. Um den Namen von Variablen der Variablenliste hinzuzufügen, die eine Aktion hervorrufen, muß man den Befehl **•AktionVar(+,VarName)** ausführen, und zwar so oft, wie es Variable zum Einfügen gibt. Umgekehrt, um ein Variable von der Liste zu entfernen, muß man den Befehl **•AktionVar(-,VarName)** ausführen, und zwar so oft, wie es Variable zum Löschen gibt. Um die Liste komplett zu löschen, führen Sie den Befehl **•AktionVar**(-) aus. Der Befehl **•AktionVar(Zeile,VarName,CoMod)** definiert eine globale Verwaltung der Aktionen, daß heißt, diese dauert von Modul zu Modul bis man auf einen desaktivierten Befehl trifft. Diese Verwaltung läßt so die CoModule ***Comod*** ausführen mit als einziger Parameter ***Zeile***, der von jeder Form sein kann. Am Ende des Comoduls nimmt die Ausführung des Modul in der Zeile wieder aufgenommen, die der Zeile folgt, wo sich die Aktion ereignete. Der Befehl **•Aktion(-)** stellt die Verwaltung der aktiven Aktionen in Schlaf, die durch den Befehl **•Aktion(+)** wider geweckt werden können. Man muß anmerken, wenn eine Aktion erscheint, die Verwaltung der Aktionen auf die Zeile oder auf das angezeigte Comodule springt und **•Evt(-),** ausführt, folglich müßte man die Verwaltung der Aktionen aufwecken (**•Evt(+)**) nachdem die behandelten Aktionen erschienen sind. Um eine spezielle Verwaltung von Aktion zu desaktivieren, reicht es, diese Verwaltung wieder mit 0 als Zeilennummer anzugeben.

BEISPIEL

```
•AktionVar(TasteKlick,B1)
•AktionVar(+,B2)
••Sprung auf den Label TasteKlick
••wenn auf die Taste B1 oder B2 geklickt wird
```

Alias: (AttenteEvt) Befehl: AktionWarte

Setzt eine Aufgabe oder ein Modul in einen Wartezustand

SYNTAX **•AktionWarte()**

BESCHREIBUNG Eine Aufgabe oder ein Modul wird in einen Wartezustand gesetzt, wobei auf eine Aktion gewartet wird.
Die Ausführung eines Moduls oder einer Aufgabe wird durch eine lokal ausgelöste Aktion aufgenommen.

ANMERKUNG Ein Modul oder eine Aufgabe mit einem **•AktionWarte()** ist in einem Wartezustand und verbraucht keine Recourcen der Maschine.

BEISPIEL

```
•Zone(Band,0,0,800,50)
•AktionVar(AmBand,Band)
•AktionWarte()
```

Alias: (ActiveExe) Befehl: AktivExtern

Aktiviert eine Anwendung

SYNTAX •**AktivExtern(*AnwName*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Aktivieren einer Anwendung, d.h. dieser Befehl übernimmt dabei die Tastatur- bzw. Maus-Aktionen

AnwName : Startname der Anwendung

BEISPIEL

```
•Extern(write,11)••startet Write
••AktivExtern(write)
••einen Text in Write schreiben
•Tastatur('Text °F13in Write°F13in mehreren Zeilen')
•Pause(5)
••speichern des Dokumentes
•Tastatur('°F470e')••Alt F e
•Pause(5)
••dem Dokument einen Namen geben
•Tastatur('Demo')•Pause(3)•Tastatur('°F13')
••Write verlassen
•Tastatur('°F470q')
```

SIEHE AUCH •**Tastatur**()

Alias: (AltTab) Befehl: AltTab

Erlauben / unterbinden der Verwendung von Alt+Tab und Ctrl+Esc

SYNTAX **•AltTab(+)** oder **•AltTab(-)**

BESCHREIBUNG Aktiviert oder desaktiviert die Möglichkeit, vom Schülermodus aus zu einer anderen Windows-Applikation mit Hilfe der Tastenkombination Alt+Tab oder Ctrl+Esc zu wechseln.

•AltTab(+) : erlaubt die Verwendung von Alt+Tab

•AltTab(-) : untersagt die Verwendung von Alt+Tab

ANMERKUNG Ohne Angabe ist der Zustand: Desaktiviert

BEISPIEL

```
•AltTab(+)
```

Alias: (Poursuite) Befehl: AnalyseFort

Fortsetzen in der Analyse

SYNTAX `•AnalyseFort()`

BESCHREIBUNG Setzt die Analyse nach der Ausführung des Kommentars fort. Das heißt, EGO setzt die Suche nach einem neuen Antwortmodell, das der Mitteilung entspricht fort, anstelle zum Interventionspunkt zurückzukehren.

Alias: (FenRep) Befehl: AntwFen

Definieren eines Antwortfensters

SYNTAX •**AntwFen(*Xh,Yh,Breite,Höhe,TFarbe,Hgrundfarbe,Mitteil, AnzZei,PWort*)**

BESCHREIBUNG Erzeugt und öffnet ein Antwortfenster mit dem Ursprung ***Xh,Yh***, der Breite ***Breite*** und der Höhe ***Höhe***. Der Text, der dort angezeigt werden wird, hat die Farbe ***Textfarbe*** und der Hintergrund des Fensters hat die Farbe ***Hgrundfarbe***.

Xh,Yh	:	Koordinaten der oberen linken Ecke des Fensters
Breite	:	Breite des Fensters
Höhe	:	Höhe des Fensters
TFarbe	:	Farbe des Textes im Fensters
Hgrundfarbe	:	Farbe des Hintergrundes des Fensters
Mitteil	:	Name einer Variablen Kette
AnzZei	:	Maximale Anzahl an Buchstaben, die erfaßt werden können
PWort	:	Zeichen verwendet für den Modus Passwort

ANMERKUNG Alle Koordinaten werden in Pixel ausgedrückt. Damit der Inhalt eines Fensters beim Öffnen nicht gelöscht wird, darf man keine Hintergrundfarbe angeben. Um ein Antwortfenster zu desaktivieren, ist der Befehl •**KomFen**() zu verwenden (ohne Angabe von Parametern zwischen den Klammern). Ein Antwortfenster kann im Textteil oder Kommentarteil jederzeit definiert werden.

Um die Mitteilung des Schülers in das Antwortfenster zurückzusenden, damit dieser sie verbessern kann, ist als vorletzter Parameter des Befehls der Name einer Variablen Kette anzugeben, die die zu editierende Mitteilung enthält.

Die Variable ***AnzZei*** ermöglicht die Beschränkung der Anzahl von Buchstaben, die vom Schüler eingegeben werden können. ***AnZei*** kann Werte zwischen 1 und 1024 annehmen.

Wenn der Parameter ***PWort*** verwendet wird, dann befindet sich das Antwortfenster im Modus PassWort und alle eingegebenen Zeichen erscheinen mit dem Zeichen ***PWort***, das für das Passwort gewählt wurde

BEISPIEL

```
•AntwFen(50,200,300,20,5,2,ewMessage)
•Eingabe()
•• Das Zeichen * wird als Passwort verwendet
•AntwFen(50,200,300,20,5,2,,,*)
•Eingabe()
```

SIEHE AUCH •**KomFen**(), •**Eingabe**()

Alias: (FenRepCmd) Befehl: AntwFenBef

Definieren die Liste von speziellen Tasten für ein Antwortfenster

SYNTAX •**AntwFenBef(Wert,Op,Ta1,...TaN)**

BESCHREIBUNG Definiert die Liste mit speziellen Tasten die das Verhalten eines Antwortfenster betreffen.

Wert : Parameter, der die Art der Tastenliste definiert ***Ta1***,...***TaN***. Der Wert dieser Parameter kann zwischen 1 und 3 betragen.
1 = Liste der Tasten, die das Antwortfensters beenden
2 = Liste der Tasten, die den Übergang zur nächsten Zeile bewirken
3 = Liste der nicht verwendbaren Tasten bei der Erfassung in einem Antwortfenster

Op: Parameter, die die ausgeführte Operation zwischen der existierenden Liste und der neuen Liste definieren. Dieser Parameter ist ein + , - oder *.
\+ = Hinzufügen einer neuen Liste zu der existierenden
\- = Löschen der Liste von der existenten Liste
* = Ersetzen der existenten Liste durch die neue Liste

Ta1...TaN : Liste von Tasten (32 Maximum)

ANMERKUNG Die Tasten ***Ta1***,...***TaN*** können beliebige Zeichen sein, oder auch Funktionstasten, repräsentiert durch °F und dem Code der Tastenfunktion, oder ein ASCII-Code, repräsentiert durch °S und dem ASCII-Code der Zeichen.
•**AntwFenBef(Zahl)** setzt die Liste ***Wert*** auf den standardmäßigen Wert.
•**AntwFenBef**() setzt alle Listen auf ihren standardmäßigen Wert.
Auf keinen Fall darf man •**AntwFenBef(1,*)** schreiben, denn dies löscht die Liste der Tasten, was einer Beendigung des Antwortfensters gleich kommt.

BEISPIEL

```
•• untersagen von Zeichen
•AntwFenBef(3,+,0,1,2,3,4,5,6,7,8,9)
•AntwFen(100,100,200,20,236,255)
•Eingabe()
•• die Standard-Liste aufrufen
•AntwFenBef(3)
```

SIEHE AUCH •**AntwFen**(), •**AntwFenOpt**(), •**Eingabe**()

Alias: (PolRep) Befehl: AntwFenFont

Laden eines Fonts (Schrift) für das Antwortfenster

SYNTAX •**AntwFenFont(*Schrift,Größe,Stärke,Italic*)**

BESCHREIBUNG Lädt eine Schrift, die im Antwortfenster des Schülers verwendet wird.

Schrift	Name der Schrift
Größe	Höhe der Buchstaben in Pixel
Stärke	Stärke der Buchstaben (0 bis 2)
Italic	1=kursiv, 0=normal
Stärke	**Strichstärke**
0	= Fein
1	= Normal
2	= Fett

ANMERKUNG Die Standardeinstellung für die Stärke ist 1 und die für Italic ist 0.

BEISPIEL

```
•AntwFenFont(System)
```

SIEHE AUCH •**Font**(), •**TasteFont**()

Alias: (FenRepOpt) Befehl: AntwFenOpt

Definiert die Beendigungsoptionen eines Antwortfenster

SYNTAX •**AntwFenOpt(1,Opt)**

BESCHREIBUNG Definiert die Beendigungskonditionen eines Antwortfensters, wenn es gesättigt ist.

1	:	Obligatorischer Parameter (geplant für zukünftige Entwicklungen)
Opt	:	Parameter zur Definition der Option zur Beendigung des Antwortfensters, unter Verwendung der Werte zwischen 0 und 3 0 = Keine automatische Beendigung bei Sättigung des Fensters (Standardzustand) 1 = Ausgabe eines Bip-Ton bei Sättigung des Fensters 2 = Automatische Beendigung bei Sättigung des Fensters 3 = Automatische Beendigung und Ausgabe eines Bip-Ton bei Sättigung des Fensters

ANMERKUNG Standardmäßig erfolgt die Beendigung eines Antwortfensters durch einen Druck auf die Taste «Enter» oder mittels definierter Tasten durch den Befehl •**AntwFenBef**().

BEISPIEL

```
•• Erfassen eines Zeichens für den Ausstieg
•AntwFenOpt(1,2)
•AntwFen(100,100,20,20,236,255,,1)
•Eingabe()
```

SIEHE AUCH •**AntwFen**(), •**AntwFenBef**(), •**Eingabe**()

Alias: (PosRep) Befehl: AntwPos

Positioniert den Cursor in einem Antwortfenster

SYNTAX •**AntwPos(*CAnfang,CEnde*)** oder •**AntwPos(*CPos*)**

BESCHREIBUNG Markiert einen Block und/oder positioniert den Cursor in einen Antwortfenster

CAnfang	:	Position Block-Anfang
CEnde	:	Position Block-Ende
CPos	:	Cursorposition

ANMERKUNG Bei der Markierung eines Blockes plaziert sich der Cursor an der Position ***CEnde***. Wenn ***CEnde*** viel kleiner als ***CAnfang*** ist, dann befindet sich der Cursor am Anfang des markierten Blocks.

BEISPIEL

```
•AntwFen(100,100,200,20,236,255,ewmessage)
•AntwPos(10)
•Eingabe()
```

SIEHE AUCH •**KomFen**()

Alias: (Tache) Befehl: Aufgabe

Ausführen einer Aufgabe

SYNTAX •**Aufgabe**(***Modulname, +/-, p1,p2,...pn***)

BESCHREIBUNG Führt ein Modul parallel zu dem Modul aus, das die Aufgabe geladen hat. Es ist möglich, bis zu 16 Aufgaben parallel zum Hauptmodul zu starten. Der Befehl •**AufgEnde**() hält die Ausführung einer Aufgabe an.

Modulname	:	Name des Moduls
+/-	:	+ oder -) globale oder lokale Aufgabe
p1,...pn	:	Zu übertragende Parameter an das Aufgaben-Modul

ANMERKUNG Eine Aufgabe kann nur ein Textteil eines Moduls sein. Die Geschwindigkeit der Ausführung der Aufgaben hängt von der Ausstattung des Computers ab.

Standardmäßig (+) ist eine Aufgabe global, das heißt sie muß mit dem Befehl •**AufgEnde**() angehalten werden.

Wenn der letzte Parameter des Befehls •Aufgabe() das Zeichen ´-´ ist, so handelt es sich um eine lokale Aufgabe, die beendet wird, sobald das Modul, das aufgerufen wurde, mit einem anderen Modul verkettet wird. Die aufgerufene Aufgabe kann die Parameter wiedererlangen, die vom vorhergehenden Modul übertragen wurden. Dies geschieht mit dem Befehl •**ModParam**(). Es können bis zu 32 Parameter übernommen werden.

BEISPIEL

```
•Aufgabe(Uhr)          ••globale Aufgabe
•Aufgabe(Anim,-)       ••lokale Aufgabe
```

SIEHE AUCH •**AufgEnde**(), •**AufgPause**(), •**AufgFort**(), •**ModParam**()

Alias: (FinTache) Befehl: AufgEnde

Beenden der Ausführung einer Aufgabe

SYNTAX •**AufgEnde(Aufgabenname)**

BESCHREIBUNG Beendet die Ausführung der Aufgabe Aufgabenname, die zuvor durch den Befehl •**Aufgabe**(Aufgabenname) gestartet wurde.

ANMERKUNG Wenn man den Befehl •**AufgEnde()** ohne Angabe des Namens der Aufgabe eingibt, so werden alle derzeit laufenden Aufgaben beendet.

BEISPIEL

```
•AufgEnde(Uhr)
```

SIEHE AUCH •**Aufgabe**(), •**AufgPause**(), •**AufgFort**()

Alias: (ContTache) Befehl: AufgFort

Fortsetzen der Ausführung einer Aufgabe, die vorübergehend angehalten wurde

SYNTAX •**AufgFort(*Aufgabenname*)**

BESCHREIBUNG Nimmt die Ausführung einer Aufgabe wieder auf, die durch den Befehl •**AufgPause**() vorübergehend angehalten wurde.

ANMERKUNG Wenn man den Befehl •**AufgFort**() verwendet ohne einen Namen anzugeben, werden alle Aufgaben fortgesetzt, die vorübergehend angehalten wurden. Um die Ausführung einer Co-Aufgabe wieder aufzunehmen, muß man den Namen der Mutter-Aufgabe verwenden.

BEISPIEL

```
•AufgFort(Uhr)
```

SIEHE AUCH •**Aufgabe**(), •**AufgPause**(), •**AufgEnde**()

Alias: (PauseTache) Befehl: AufgPause

Vorübergehendes Anhalten einer Aufgabe (schlafend setzen)

SYNTAX •**AufgPause**(***AufgName***)

BESCHREIBUNG Unterbricht die Ausführung der Aufgabe ***AufgName***, die zuvor mit dem Befehl •**Aufgabe**(AufgName) aktiviert wurde.

ANMERKUNG Wenn man in einem Modul den Befehl •**AufgPause**() ohne Namen der Aufgabe angibt, so werden alle Aufgaben, die gerade ausgeführt werden, gestoppt.

Wenn der Befehl •**AufgPause**() innerhalb einer Aufgabe angegeben wird ohne den Namen der Aufgabe anzugeben, so werden alle aktuellen Aufgaben ausgesetzt, auch die Ausführung des Moduls, ausgenommen die Aufgabe, in der der Befehl vorkommt. Um eine CoAufgabe zu setzen, muß man den Namen der Ausgangs-Aufgabe durchlaufen.

BEISPIEL

```
•AufgPause(Uhr)
```

SIEHE AUCH •**Aufgabe**(), •**AufgPause**(), •**AufgFort**() •**CoAufg**(),

Alias: (Justif) Befehl: Ausrichten

Ausrichten des Textes

SYNTAX •**Ausrichten(*Art*)**

BESCHREIBUNG Definiert die Art der Ausrichtung des Textes in einem Fenster.

Art : Art der Ausrichtung (zwischen 0 und 3)

Art	**Ausrichtung**
0	= Linksbündig (Standardeinstellung)
1	= Rechtsbündig
2	= Zentriert
3	= Blocksatz

BEISPIEL

```
•Ausrichten(2)    ••Zentrieren
•KomFen(100,100,40,20,237,241)
DIESER TEXT IST ZENTRIERT.
```

SIEHE AUCH •**KomFen**()

Alias: (CPR) Befehl: BedAM

Bedingung in einem Antwortmodell

SYNTAX •**BedAM([*Ausdruck*])**

BESCHREIBUNG Dieser Befehl ermöglicht die Integration von numerischen und/oder alphanumerischen Bedingungen in einem Antwortmodell. Die Bedingung muss immer zwischen Klammern geschrieben werden.

Ausdruck : Bedingung

ANMERKUNG Der Befehl •**BedAM()** kann nur in einem Antwortmodell verwendet werden.

Das Äquivalent für den Text- oder Kommentarteil stellt der Befehl •**Wenn**() dar.

BEISPIEL

```
•BedAM([Zähler=10 & C>2])
```

SIEHE AUCH •**Wenn**() und die Operatoren und Funktionen des Rechners

Alias: (LHDiv) Befehl: BHDiv

Definieren der Divisoren für die Anzeige von verschiedenen Bildschirmwippen

SYNTAX •**BHDiv(*Breite_div,Höhe_div*)**

BESCHREIBUNG Definiert die Breite und die Höhe der Grafikblocks, wenn die Anzeige als Mosaik oder als Spirale erfolgen soll.

Breite_div : Die Breite eines Grafikblocks ergibt sich aus der Breite des Bildschirmes dividiert durch ***Breite_div***.

Höhe_div : Die Höhe eines Grafikblocks ergibt sich aus der Höhe des Bildschirmes dividiert durch ***Höhe_div***.

ANMERKUNG Der Standardwert der Divisoren ist mit 30 festgelegt. Der Maximalwert der Divisoren ist 100.

BEISPIEL

```
•BHDiv(80,40)
•Objekt(Foto,120,200,,514)    ••Spirale
```

SIEHE AUCH •**Bild**(), •**Objekt**()

Alias: (Image) Befehl: Bild

Anzeigen eines Bildes

SYNTAX •**Bild(*Name,Alias,Xdh,Ydh,Xdb,Ydb,Modus,Op,Xsh,Ysh,Xsb,Ysb*)**

BESCHREIBUNG Ermöglicht die Anzeige, Manipulation und Unterteilung von Grafiken.

Name	:	Name der anzuzeigenden Grafik
Alias	:	Name des gespeicherten Hintergrundes
Xdh,Ydh		
Xdb,Ydb	:	Koordinaten der Diagonale der Oberfläche, wo die Grafik angezeigt wird
Modus	:	Anzeigemodus der Grafik (0 bis 14)
Op	:	Operationen, die mit dem Bild ausgeführt werden sollen
Xsh,Ysh		
Xsb,Ysb	:	Koordinaten der Diagonale der Oberfläche der Grafik, die angezeigt werden soll

Modus **Anzeigemodus**

0	=	negativ, alle Farben invers
1	=	normal (Standardeinstellung)
2	=	logisches ODER zwischen Bild und Hintergrund
3	=	logisches UND zwischen Bild und Hintergrund
4	=	logisches XODER zwischen Bild und Hintergrund
5	=	logisches UND zwischen negativem Bild und Hintergrund
6	=	ogisches ODER zwischen negativem Bild und Hintergrund
7	=	Quelle UND Ziel
8	=	NON Quelle ODER Ziel
9	=	Normal
10	=	(NON Quelle ODER Ziel) ODER Ziel
11	=	Quelle XODER Ziel
12	=	Invers
13	=	zeichnet ein schwarzes Rechteck
14	=	zeichnet ein weißes Rechteck

OP **Operation**

0	=	Lädt eine Grafik vorab in den Speicher
1	=	Aktivierung der Palette
2	=	Darstellung ohne die Palette
4	=	Darstellung mit der Maske
8	=	Darstellung mit Speicherung des Hintergrundes
16	=	Darstellung der Maske des Bildes
32	=	Reduktion auf 236 Farben der Farbpalette des Bildes
128	=	Darstellung des Bildes in Mosaikform
256	=	Darstellung als nach innen verlaufende Spirale
512	=	Darstellung als nach außen verlaufende Spirale
1024	=	Darstellung mit Vorhang nach links
2048	=	Darstellung mit Vorhang nach rechts
4096	=	Darstellung mit Vorhang nach oben
8192	=	Darstellung mit Vorhang nach unten
16384	=	Darstellung mit Zeilen nach Zufall

ANMERKUNG

Der Wert ***OP*** (Operation) ist das Resultat der Summe der verschiedensten Optionen der Darstellung. Um zum Beispiel ein Bild mit seiner Palette, in Mosaikform, mit Speicherung des Hintergrundes darzustellen, ergibt sich für ***OP*** folgender Wert: **2+1+128+8=139**.

Für die Darstellung mit horizontalem und vertikalem Vorhang legen ***Breite_div*** und ***Höhe_div*** im Befehl •**BHDiv**() die Anzahl der Darstellungsetappen fest.

Auch die Darstellung von Zufallslinien ist abhängig von ***Breite_div*** und ***Höhe_div*** im Befehl •**BHDiv**(). Wenn ***Breite_div*** größer ist als ***Höhe_div***, dann sind die Linien horizontal und die Stärke der Linien ergibt sich aus der Differenz von ***Breite_div*** und ***Höhe_div***. Wenn ***Höhe_div*** größer ist als ***Breite_div***, dann sind die Linien vertikal. Die Option OP = 32 erlaubt das Anpassen eines Bildes an die reduzierte Palette mit 236 Farben, wobei wichtig ist, daß die 20 System-Farben (236 bis 255) die Bildwiedergabe überlagern.

Wenn man eine Grafik anzeigen will, ohne ihre Größe zu verändern, müssen die Koordinaten ***Xdb*** und ***Ydb*** nicht angeben werden.

BEISPIEL

```
••Einfache Anzeige einer Grafik mit Palette
•Bild(Grafik.PCX,,0,0,0,0,,3)
••Einfache Anzeige einer Grafik mit reduzierter Palette
•Bild(Grafik.PCX,,0,0,0,0,,35)
••Anzeige einer Grafik mit Speicherung des Hintergrunds
•Bild(IMA2.PCX,Hintergrund,100,200,,,,10)
••Wiederherstellen des vorher gespeicherten Hintergrunds
•Bild(Hintergrund,,100,200,,,,2)
••Speicherung eines Bildschirmteiles
•Bild(,Bildschirm,200,200,350,300,,10)
••Kopie eines Bildschirmausschnittes mit den Ursprungs
••Koordinaten 300,100,400,400 nach 10,10,110,130
•Bild(,,10,10,110,130,,2,300,100,400,400)
••Anzeige eines vergrößerten Grafikausschnittes
•Bild(Grafik.PCX,,100,100,400,400,,2,100,100,200,200)
```

SIEHE AUCH

•**Objekt**(), •**GrafikFormat**(), •**BHDiv**(), •**BildDim**(), •**BildEntf**(), •**BildMax**(), •**Transp**()

Alias: (AdapteImg) Befehl: BildAnPal

Anpassen eines Bildes an eine vorgegebene Palette

SYNTAX •**BildAnPal(*BildnameO,BildnameA,PalNum*)**

BESCHREIBUNG Erstellt ein Bild ***BildnameA***, angepaßt an die virtuelle oder physische Palette ***PalNum*** des Bildes ***BildnameO***.

BildnameO	:	Name des Original-Bildes
BildnameA	:	Name des angepaßten Bildes
PalNum	:	Nummer der Pallette, an die angepaßt wird (0 bis 16)

ANMERKUNG Das Originalbild ***BildnameO*** muß vor der Verwendung des Befehls •**BildAnPal**() bereits geladen sein. Ohne Angabe hat die anzupassende Palette die Nummer 0. Dieser Befehl wirkt sich nur im Graphikmodus mit 256 Farben aus.

BEISPIEL

```
•Objekt(PalRef.pcx,,,,1)    ••Lädt eine Palette
•Objekt(Foto.pcx,,,,0)    ••Vorheriges Laden des Bildes
•BildAnPal(Foto.pcx,FotoAn.pcx,0)  ••Angepaßt an Palette0
•Objekt(FotoAn.pcx)    ••Anzeige des angepaßten Bildes
```

SIEHE AUCH •**Objekt**(), •**Bild**(), •**PalKopie**(), •**PalOp**(), •**RGBWechsel**(), •**Farbskala**()

Alias: (ImageLH) Befehl: BildBH

Anzeigen eines Bildes

SYNTAX •**BildBH(Name,Alias,Xdh,Ydh,Bd,Hd,Modus,Op,Xsh,Ysh,Bs,Hs)**

BESCHREIBUNG Ermöglicht die Anzeige, Manipulation und Unterteilung von Grafiken.

Name	:	Name der anzuzeigenden Grafik
Alias	:	Name des gespeicherten Hintergrundes
Xdh,Ydh	:	Koordinaten der linken oberen Ecke der Fläche, wo die Grafik angezeigt wird
Bd,Hd	:	Breite und Höhe der Fläche, in welcher das Bild angezeigt wird
Modus	:	Anzeigemodus der Grafik (0 bis 14)
Op	:	Operationen, die mit dem Bild ausgeführt werden sollen
Xsh,Ysh	:	Koordinaten der linken oberen Ecke der Fläche der Grafik, die angezeigt werden soll
Bs,Hs	:	Breite und Höhe der Fläche des Bildes, welches angezeigt werden soll

ANMERKUNG Für weitere Details über Modus und Operationen siehe Befehl •**Bild**()

BEISPIEL

```
•BildBH(Bild.PCX,,100,100,400,400,,2,100,100,100,100)
```

SIEHE AUCH •**Bild**(), •**Objekt**(), •**GrafikFormat**(), •**BHDiv**(), •**BildDim**(), •**BildEntf**(), •**BildMax**(), •**Transp**()

Alias: (ImgDim) Befehl: BildDim

Ermitteln der Abmessungen eines Bildes

SYNTAX •**BildDim(*Grafikname,Breite,Höhe*)**

BESCHREIBUNG Ermittelt die Breite ***Breite*** und die Höhe ***Höhe*** einer Grafik.

Grafikname	:	Name der Grafik
Breite	:	Name der numerischen Variablen, in der die Breite der Grafik gespeichert wird
Höhe	:	Name der numerischen Variablen, in der die Höhe der Grafik gespeichert wird

ANMERKUNG Die Grafik muß zuvor mit dem Befehl •**Bild**() oder •**Objekt**() geladen werden.

BEISPIEL

```
•Objekt(Photo.PCX,200,300,,0)
•BildDim(Photo.PCX,Breite,Höhe)
```

SIEHE AUCH •**Objekt**(), •**Bild**(), •**GrafikFormat**()

Alias: (LibImg) Befehl: BildEntf

Entfernen eines Bildes aus dem Arbeitsspeicher

SYNTAX •**BildEntf(*Bildname*)**

BESCHREIBUNG Entfernt das Bild ***Bildname*** aus dem Arbeitsspeicher.

Bildname : Name des Bildes

ANMERKUNG Wenn der Befehl ohne Angabe eines Bildnamens verwendet wird, werden alle resistent im Speicher vorhandenen Bilder entfernt.

Man kann auch eine Liste von zu löschenden Bildern angeben, jeder Name muß durch einen Beistrich getrennt sein.

BEISPIEL

```
•BildEntf(FOTO.PCX,E:\BILDER\FISCH.TIF)
```

SIEHE AUCH •**Objekt**(), •**Bild**(), •**Grafikformat**(), •**BildMax**(), •**BildEntfA**()

Alias: (LibImgSauf) Befehl: BildEntfA

Entfernen aller Bilder aus dem Arbeitsspeicher, bis auf jene, die aufgelistet sind

SYNTAX •**BildEntfA(*Bildname1, ... , BildnameN*)**

BESCHREIBUNG Entfernt alle Bilder aus dem Arbeitsspeicher, bis auf jene, die in der Liste ***Bildname1*, ... *BildnameN*** aufscheinen.

Bildname1...N : Namen der Bilder, die im Arbeitsspeicher verbleiben sollen

ANMERKUNG Wenn der Befehl ohne Angabe eines Bildnamens verwendet wird, werden alle resistent im Speicher vorhandenen Bilder entfernt.

BEISPIEL

```
•BildEntfA(FOTO.PCX,E:\BILDER\FISCH.TIF)
```

SIEHE AUCH •**Objekt**(), •**Bild**(), •**Grafikformat**(), •**BildMax**(), •**BildEntf**()

Alias: (MaxImg) Befehl: BildMax

Festlegen der maximalen Anzahl an geladenen Bildern

SYNTAX •**BildMax(*Anzahl*)**

BESCHREIBUNG Legt die maximale Anzahl an Bildern fest, die EGO gleichzeitig in den Arbeitsspeicher laden kann.

Anzahl : maximale Anzahl an Bildern im Arbeitsspeicher

ANMERKUNG Sobald EGO den Befehl •**Bild**() oder •**Objekt**() antrifft, lädt es das Bild in den Arbeitsspeicher und speichert den Namen.

Daher wird das Bild bei einem neuerlichen Aufruf nicht neu von der Festplatte geladen. Da dieser Prozeß viel Arbeitsspeicher beansprucht, ist es von Nutzen, die Anzahl der im Speicher residenten Grafiken zu beschränken.

EGO verwaltet einen Grafikstapel und wenn dieser voll ist, so ersetzt es die zuerst geladenen Grafik durch die neue.

Die Standardeinstellung für die maximale Anzahl ist 64. Die Zahl Anzahl kann Werte zwischen 1 und 300 annehmen.

BEISPIEL

```
•BildMax(5)
```

SIEHE AUCH •**Bild**(), •**Objekt**()

Alias: (NomImg) Befehl: BildName

Ermitteln des Namens eines im Arbeitsspeicher geladenen Bildes

SYNTAX •**BildName(*Nummer,BName*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln des Namens eines im Arbeitsspeicher geladenen Bildes.

Nummer : Nummer des Bildes, dessen Namen man ermitteln möchte

BName : Name der Zeichenketten-Variablen, welcher der Name des Bildes zugewiesen wird

BEISPIEL

```
••Ermittelt die Anzahl der geladenen Bilder
•BildZahl(AnzBild)
•Wenn([AnzBild]=0)•Ende()
[i:=1]
••Ermittelt den Namen des Bildes und zeigt ihn an
•BildName([i],Name)
[i:=i+1]
•Schleife([AnzBild-1],=-2)
```

SIEHE AUCH •**BildMax**(), •**BildZahl**()

Alias: (NbImg) Befehl: BildZahl

Ermitteln der Anzahl der im Arbeitsspeicher geladenen Bilder

SYNTAX •**BildZahl(*Zahl*)**

BESCHREIBUNG Dieser Befehl erlaubt das Ermitteln der Anzahl der im Arbeitsspeicher geladenen Bilder.

Zahl : Name der numerischen Variablen, welcher die Anzahl der geladenen Bilder zugewiesen wird

BEISPIEL

```
•BildZahl(AnzBild
```

SIEHE AUCH •**BildMax**(), •**BildName**()

Alias: (Arc) Befehl: Bogen

Zeichnen eines Bogens

SYNTAX •**Bogen(*Xc,Yc,Winkel1,Winkel2,Rx,Ry*)**

BESCHREIBUNG Zeichnet einen Ellipsen- oder Kreisbogen mit dem Mittelpunkt ***Xc,Yc***, mit dem ausgehenden Winkel ***Winkel1*** und dem ankommenden Winkel ***Winkel2*** und den Radien ***Rx,Ry***.

Xc,Yc	:	Koordinaten des Zentrums des Bogens
Winkel1,Winkel2	:	Winkel des Bogens
Rx,Ry	:	Radius des Bogens

ANMERKUNG Die Winkel ***Winkel1*** und ***Winkel2*** werden in Zehntel-Grad ausgedrückt.

Um einen Kreisbogen zu zeichnen, genügt die Angabe nur eines Radius.

Die Zeichnung erfolgt in der Farbe, im Modus und im Stil des aktuellen Pinsels.

BEISPIEL

```
•Bogen(320,200,1200,2350,50,30)    ••Ellipsenbogen
•Bogen(200,150,900,1800,120)       ••Kreisbogen
```

SIEHE AUCH •**Kreis**(), •**EllipseVoll**(), •**Ellipse**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Ecran) Befehl: Bsch

Auswählen des aktiven Bildschirms

SYNTAX •**Bsch(*Nummer*)**

BESCHREIBUNG Auswahl des aktiven Bildschirms, d.h. des Bildschirms, auf dem die Befehle (z.B. Grafiken und Texte) ausgeführt werden sollen.

Halten wir nochmals fest, daß Bildschirm 0 der physische Bildschirm des aktiven Fensters ist.

BEISPIEL

```
•VBschDef(200,200)     ••definiert die Auflösung des
                       ••virtuellen Bildschirms
•Bsch(2)               ••aktiviert den virtuellen Bildschirm2
```

SIEHE AUCH •**VBschDef**(), •**BschLösch**()

Alias: (Imp) Befehl: BschDruck

Ausdrucken eines Bildschirmbereiches

SYNTAX •**BschDruck(*Xsh,Ysh,Xsb,Ysb,Xdh,Ydh,Xdb,Ydb,Einheit*)**

BESCHREIBUNG Druckt einen Bildschirmbereich auf dem aktuellen Drucker aus.

Xsh,Ysh

Xsb,Ysb : Koordinaten der Diagonale des Bildschirm-bereiches der ausgedruckt werden soll

Xdh,Ydh

Xdb,Ydb : Koordinaten der Diagonale der Fläche, wo der Bildschirminhalt ausgedruckt werden soll

Einheit : Einheit (0 = Zentimeter oder 1 = Zoll) der Dimension der Ausdruckfläche

ANMERKUNG Die Koordinaten der Ausdruckfläche werden in 1/100stel Einheiten ausgedrückt.

Die Standardeinheit ist Zentimeter.

BEISPIEL

```
•BschDruck(100,100,200,300,0,0,1000,1000,0)
```

SIEHE AUCH •**Bsch**(), •**BschDruckW**()

Alias: (ImpEcr) Befehl: BschDruckW

Erlauben / unterbinden der Verwendung der Windows-Zwischenablage

SYNTAX •**BschDruckW(+/-)**

BESCHREIBUNG Aktiviert bzw. desaktiviert die Möglichkeit der Verwendung der Windows-Zwischenablage im Schülermodus.

- ***•BschDruckW(+)*** : Erlaubt die Benutzung der Windows-Zwischenablage
- ***•BschDruckW(-)*** : Unterbindet die Benutzung der Windows-Zwischenablage

ANMERKUNG Die Standardeineinstellung ist: Aktiviert

BEISPIEL

```
•BschDruckW(-)
```

Alias: (CopEcran) Befehl: BschKopie

Kopie zwischen virtuellem und physischem Bildschirm

SYNTAX •**BschKopie**(***BschQ,BschZ,Xsh,Ysh,Xsb,Ysb,Xdh,Ydh,Xdb,Ydb,Mod,Opt***)

BESCHREIBUNG Ermöglicht das Kopieren des gesamten physischen oder virtuellen Bildschirms oder eines Teils davon in den jeweils anderen.

BschQ	:	Quellbildschirm
BschZ	:	Zielbildschirm
Xsh,Ysh		
Xsb,Ysb	:	Koordinaten der Oberflächendiagonale des Quellbildschirms
Xdh,Ydh		
Xdb,Ydb	:	Koordinaten der Oberflächendiagonale des Zielbildschirms, auf den die Oberfläche des Quellbildschirms kopiert wird
Mod	:	Anzeigemodus (ohne Angabe: Kopie). Es handelt sich um dieselben Modi wie bei •**Bild**()
Opt	:	Option mit der Darstellung als Maske (transparent)
4	=	Bildschirm-Kopie mit Transparenz (wie bei •**Bild**())

ANMERKUNG Die Koordinaten ***Xdb,Ydb*** sind nur notwendig, wenn man die Größe der Darstellung am Zielbildschirm verändern will. Der Parameter ***OP*** ermöglicht das Erstellen einer Bildschirmkopie mit Transparenz. Die transparente Farbe wird mit Hilfe des Befehls •**BschTransp**() festgelegt

BEISPIEL

```
•BschKopie(1,0,100,100,50,70,400,300)
```

SIEHE AUCH •**VBschDef**(), •**Bsch**(), •**VBschLösch**(), •**WipeHor**()

Alias: (CopEcranLH) Befehl: BschKopieBH

Kopie zwischen virtuellem und physischem Bildschirm

SYNTAX •**BschKopieBH(*BschQ,BschZ,Xsh,Ysh,Bq,Hq,Xdh,Ydh,Bz,Hz,Mod*)**

BESCHREIBUNG Ermöglicht das Kopieren des gesamten physischen oder virtuellen Bildschirms oder eines Teils davon in den jeweils anderen.

BschQ	:	Quellbildschirm
BschZ	:	Zielbildschirm
Xsh,Ysh	:	Koordinaten der linken oberen Ecke der zu kopierenden Fläche des Quellbildschirms
Bq,Hq	:	Breite und Höhe der zu kopierenden Fläche
Xdh,Ydh	:	Koordinaten der linken oberen Ecke der Fläche des Zielbildschirms, auf den die Fläche des Quellbildschirms kopiert wird
Bz,Hz	:	Breite und Höhe der Fläche des Zielbildschirms
Modus	:	Anzeigemodus (ohne Angabe: Kopie), es handelt sich um dieselben Modi wie bei •**Bild**()

ANMERKUNG Die Breite ***Bz*** und die Höhe ***Hz*** sind nur notwendig, wenn man die Größe der Darstellung am Zielbildschirm verändern will.

BEISPIEL

```
•BschKopieBH(1,0,100,100,50,70,400,300)
```

SIEHE AUCH •**BschKopie**() •**VBschDef**(), •**Bsch**(), •**VBschLösch**(), •**WipeHor**()

Alias: (EcranListe) Befehl: BschListe

Ermittelt die Liste der virtuellen Bildschirme

SYNTAX •**BschListe(*VarName*)**

BESCHREIBUNG Ermittelt die Liste der existierenden virtuellen Bildschirme

VarName : Name der Zeichenkettenvariablen, die die Nummern der existierenden virtuellen Bildschirme enthält

ANMERKUNG Die Nummern der existierenden virtuellen Bildschirme sind in der Zeichenkette ***VarName*** enthalten, welche das Leerzeichen als Trennzeichen hat

BEISPIEL

```
•BschListe(Liste)
```

SIEHE AUCH •**VBschDef**(), •**Bsch**(), •**BschLösch**()

Alias: (Eff) Befehl: BschLösch

Löschen des Bildschirms

SYNTAX

- **BschLösch()** oder
- **BschLösch(*Index*)** oder
- **BschLösch(*R,G,B*)**

BESCHREIBUNG Löscht den Bildschirm mit der angegebenen Farbe.

- ***BschLösch()*** : Löscht den Bildschirm mit der Farbe der Füllfarbe
- ***BschLösch(Index)***: Löscht den Bildschirm mit der Farbe Index
- ***BschLösch(R,G,B)***: Löscht den Bildschirm mit der Farbe aus den Komponenten R,G,B

BEISPIEL

```
••löscht den Bildschirm mit der Farbe 10
•BschLösch(10)
••löscht den Bildschirm mit der Farbe grau
•BschLösch(128,128,128)
```

SIEHE AUCH •**Füllfarbe**()

Alias: (TranspEcran) Befehl: BschTransp

Erstellen einer Maske eines virtuellen Bildschirms.

SYNTAX •**Transp(*BschNr,Farbe*)**

BESCHREIBUNG Erstellt eine Maske des virtuellen Bildschirms ***BschNr*** mit ***Farbe*** als Transparenzfarbe

BschNr : Nummer des virtuellen Bildschirms

Farbe : Transparente Farbe

ANMERKUNG Die Erstellung einer Maske ist notwendig, wenn man einen Bereich eines virtuellen Bildschirms auf einen Hintergrund mit einer oder mehreren transparenten Farben kopieren möchte:

BEISPIEL

```
•Transp(1,0)
•BschKopie(1,0,0,0,200,200,100,100,,,,4)
```

SIEHE AUCH •**BschKopie()**

Alias: (Capt) Befehl: Capt

Speichern eines Bildschirmbereiches

SYNTAX •**Capt(*Dateiname,Xh,Yh,Xb,Yb*)**

BESCHREIBUNG Erfaßt einen Bereich am aktiven Bildschirm und speichert diesen in eine Datei.

Dateiname	:	Name der Bild-Datei
Xh,Yh,Xb,Yb	:	Koordinaten der Diagonale des Bereiches, der erfaßt und gespeichert werden soll

ANMERKUNG Das Format der Grafikdatei ist jenes, welches mit dem Befehl •**GrafikFormat**() ausgewählt wurde oder das durch die Datei-erweiterung beim Dateinamen festgelegt wird.

BEISPIEL

```
•Capt(Bild.BMP,100,100,200,300)
```

SIEHE AUCH •**CaptBH**(), •**Grafikformat**(), •**Bsch**()

Alias: (CaptLH) Befehl: CaptBH

Speichern eines Bereiches des Bildschirms

SYNTAX •**CaptBH(*Dateiname,Xh,Yh,B,H*)**

BESCHREIBUNG Erfaßt einen Bereich am aktiven Bildschirm und speichert diesen in eine Datei.

Dateiname	:	Name der Bild-Datei
Xh,Yh	:	Koordinaten der linken oberen Ecke jenes Bereiches, der erfaßt und gespeichert werden soll
B,H	:	Breite und Höhe des zu speichernden Bereiches

ANMERKUNG Das Format der Grafikdatei ist jenes, welches mit dem Befehl •**GrafikFormat**() ausgewählt wurde oder das durch die Dateierweiterung beim Dateinamen festgelegt wird.

BEISPIEL

```
•CaptBH(Bild.BMP,100,100,200,300)
```

SIEHE AUCH •**Capt**(), •**Grafikformat**(), •**Bsch**()

Alias: (ChaFic) Befehl: ChaDatei

Definieren einer Zeichenketten-Datei

SYNTAX •**ChaDatei(*Dateiname,AnzZei*)**

BESCHREIBUNG Definieren einer Zeichenkettendatei ***Dateiname***, deren Zeichenkette maximal ***AnzZei*** -Zeichen enthalten kann.

Dateiname : Name der Datei

AnzZei : Maximale Anzahl an Zeichen pro Zeichenkette

ANMERKUNG Wird nichts anderes angegeben, so wird die maximale Anzahl an Zeichen pro Zeichenkette beim ersten Schreiben in eine Datei mit 256 festgelegt. Die Zahl ***AnzZei*** kann Werte zwischen 1 und 1024 annehmen.

BEISPIEL

```
•ChaDatei(Name,40)
[Name{1}:=´3P Informatik´]
```

Alias: (ChaFicIns) Befehl: ChaDateiEinf

Fügt eine Zeichenkette in einer Zeichenketten-Datei ein

SYNTAX •**ChaDateiEinf(*ZKetteDatei,ZKetteNr,ZKetteName*)**

BESCHREIBUNG Fügt eine Zeichenkette ***ZKetteName*** in einer Zeichenketten-Datei ***ZKetteDatei*** an der Position ***ZKetteNr*** ein

ZKetteDatei	:	Name der Zeichenketten-Datei
ZKetteNr	:	Gibt an, wo die Zeichenkette eingefügt werden soll
ZKetteName	:	Name der Variablen, die die einzufügende Zeichenkette enthält

ANMERKUNG Dieser Befehl sollte mit Umsicht verwendet werden

BEISPIEL

```
[Neu:='Helga/Gerlinde/']
•ChaDateiEinf(Name,33,Neu)
```

SIEHE AUCH •**ChaDateiLösch**()

Alias: (ChaFicSup) Befehl: ChaDateiLösch

Löscht eine Zeichenkette in einer Zeichenketten-Datei

SYNTAX •**ChaDateiLösch(*ZKetteDatei,ZKetteNr*)**

BESCHREIBUNG Löscht in einer Zeichenketten-Datei ***ZKetteDatei*** eine Zeichenkette an der Position ***ZKetteNr***

ZKetteDatei : Name der Zeichenketten-Datei

ZKetteNr : Gibt die Nummer der zu löschenden Zeichenkette an

ANMERKUNG Dieser Befehl sollte mit Umsicht verwendet werden

BEISPIEL

```
•ChaDateiLösch(Name,33)
```

SIEHE AUCH •**ChaDateiEinf**()

Alias: (ChaRemp) Befehl: ChaErsetzen

Ersetzen jedes Vorkommens eines Wortes in einer Liste

SYNTAX •**ChaErsetzen(*ListennameC1,Wort,NeuesWort,Trennz*)**

BESCHREIBUNG Ersetzt jedes Vorkommen von ***Wort*** durch ***NeuesWort*** in der Variablen Liste ***ListennameC1***.

ListennameC1	:	Name der Variablen Liste C1
Wort	:	Element, das in der Liste C1 gesucht wird
NeuesWort	:	Element, das das Element Wort in der Liste C1 ersetzt
Trennz	:	Trennzeichen zwischen den Worten IMMER zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, so erfolgt die Operation auf Basis des Leerzeichens.

BEISPIEL

```
[Formel:='a-+(b+c)-+d']
•ChaErsetzen(Formel,'-+','-')
[Formel]
```

Alias: (ChaExclus) Befehl: ChaExclus

Heraussuchen derjenigen Elementen aus der Liste C1, die nicht in der Liste C2 enthalten sind

SYNTAX •**ChaExclus(*ListennameC1,ListennameC2,ListennameC3,Trennz*)**

BESCHREIBUNG Sucht aus den Elementen der Liste C1 jene heraus, die nicht in der Liste C2 enthalten sind und schreibt sie in die Liste C3, und zwar mit dem Trennzeichen, das den Listen C1 und C2 gemeinsam ist. Sofern kein Trennzeichen angegeben wird - gilt das Leerzeichen als Trennzeichen.

ListennameC1	:	Name der Variablen Liste C1
ListennameC2	:	Name der Variablen Liste C2
ListennameC3	:	Name der Variablen Liste C3
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, dann erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Die Listen ***C1*** und ***C2*** müssen dasselbe Trennzeichen haben.

•ChaExclus(ListennameC1,ListennameC2,ListennameC3,Trennz*)*

entspricht NICHT

•ChaExclus(ListennameC2,ListennameC1,ListennameC3,Trennz)

BEISPIEL

```
[Pilze:=´Morchel/Pfifferling/Knollenblätterpilz/Parasol/´]
[Tödlich:=´Fliegenpilz/Satanspilz/Knollenblätterpilz/´]
•ChaExclus(Pilze,Tödlich,Genießbar,´/´]
••Nach der Ausführung dieser Zeilen wird die Liste
••"Genießbar" folgendes enthalten:
••Morchel/Pfifferling/Parasol
```

Alias: (ChaInters) Befehl: ChaIdent1

Heraussuchen der gemeinsamen Elemente aus den Listen C1 und C2, die auch an der gleichen Position stehen

SYNTAX •**ChaIdent1(*ListennameC1,ListennameC2,ListennameC3,Trennz*)**

BESCHREIBUNG Sucht aus den Listen C1 und C2 die gemeinsamen Elemente, die sich auch an der gleichen Position befinden und erzeugt die Liste C3, die ebensoviele Elemente enthält wie die Liste C1 oder C2. Sie enthält eine Folge von 1 oder 0, je nachdem ob die Elemente der Listen C1 und C2 an identischer Stelle sind oder nicht.

ListennameC1	:	Name der Variablen Liste C1
ListennameC2	:	Name der Variablen Liste C2
ListennameC3	:	Name der Variablen Liste C3
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wurde, erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Die Listen ***C1*** und ***C2*** müssen das gleiche Trennzeichen haben. Die erzeugte Liste ***C3*** hat das Leerzeichen als Trennzeichen.

BEISPIEL

```
[Wort1:='Quelle']
[Wort2:='Qualle']
•ChaIdent1(Wort1,Wort2,Resultat)
••Nach der Ausführung dieser Zeilen wird die
••Zeichenkette als Resultat folgendes enthalten:
[Resultat]
•• 1 1 0 1 1 1 wird dargestellt
••wobei das Leerzeichen das Trennzeichen darstellt
```

Alias: (ChaInter2) Befehl: ChaIdent2

Heraussuchen der gemeinsamen Elemente aus den Listen C1 und C2

SYNTAX •**ChaIdent2(*ListennameC1,ListennameC2,ListennameC3,Trennz*)**

BESCHREIBUNG Sucht aus den Listen C1 und C2 die gemeinsamen Elemente heraus und schreibt sie in die Liste C3 mit dem Trennzeichen, das C1 und C2 gemeinsam ist oder - wenn kein Trennzeichen angegeben wird - mit dem Leerzeichen.

ListennameC1	:	Name der Variablen Liste C1
ListennameC2	:	Name der Variablen Liste C2
ListennameC3	:	Name der Variablen Liste C3
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, dann erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Die Listen ***C1*** und ***C2*** müssen dasselbe Trennzeichen haben.

BEISPIEL

```
[Pilz:=´Morchel/Pfifferling/Knollenblätterpilz/Parasol/´]
[Tödlich:=´Fliegenpilz/Satanspilz/Knollenblätterpilz/´]
•ChaIdent2(Pilz,Tödlich,Ungenießbar,´/´]
••Nach der Ausführung dieser Zeilen wird die
••Liste "Ungenießbar" folgendes enthalten:
••Knollenblätterpilz/
```

Alias: (ChaSup) Befehl: ChaLösch

Löschen eines Wortes in einer Variablen Liste

SYNTAX •**ChaLösch(*ListennameC1,Wort,Trennz*)**

BESCHREIBUNG Unterdrückt jedes Vorkommen des Elementes Wort in der Variablen Liste C1

ListennameC1	:	Name der Variablen Liste C1
Wort	:	Element, das in C1 unterdrückt werden soll
Trennz	:	Trennzeichen zwischen den Worten immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf Basis des Leerzeichens.

BEISPIEL

```
[Farbe:='rot/schwarz/blau/schwarz/grün/braun/schwarz/']
•ChaLösch(Farbe,'schwarz','/')
```

Alias: (ChaTri) Befehl: ChaOrd

Ordnen einer Zeichenkette oder einer Zeichenketten-Datei

SYNTAX

- **ChaOrd(*ChaQ,ChaZ,Trennz,Richtung,Klein/Groß*)** oder
- **ChaOrd(*ChaD{A},ChaD{E},WortNum,Trennz,Richtung,Klein/Groß*)**

BESCHREIBUNG Ordnet die Wörter oder Zeichen einer Zeichenkette im Arbeitsspeicher oder eine Zeichenketten-Datei nach speziellen Optionen.

ChaQ	:	Name der zu ordnenden Zeichenkette
ChaZ	:	Name der geordneten Zeichenkette
ODER		
ChaD{A}	:	Nummer der Zeichenkette der Datei, ab welcher das Ordnen erfolgen soll
ChaD{E}	:	Nummer der Zeichenkette der Datei, bis zu welcher das Ordnen erfolgen soll
WortNum	:	Nummer des Wortes der Zeichenkettendatei, in dessen Abhängigkeit das Ordnen erfolgen soll
UND		
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf der Ebene von Zeichen im Fall der Zeichenkette im Arbeitsspeicher und auf der Ebene der Zeichenketten im Fall einer Datei
Richtung		Art des Ordnens 0 = Aufsteigend (Grundeinstellung) 1 = Absteigend
Klein/Groß	:	Unterscheidung von Klein- und Großbuchstaben 0 = Keine Unterscheidung (Grundeinstellung) 1 = Unterscheidung

ANMERKUNG Alle Parameter, außer dem Namen der Quell-Zeichenkette und der ersten Zeichenkette der zu ordnenden Datei sind optional. Wenn man im Fall des Ordnens einer Zeichenkette im Arbeitsspeicher darauf vergißt eine Zielzeichenkette anzugeben, erhält die Quell-Zeichenkette das Resultat des Ordnens. Wenn man im Fall einer Datei darauf vergißt die Nummer der Zeichenkette anzugeben, erhält man das Ergebnis am Ende der Datei.

BEISPIEL

```
[Name:='Helga/Albert/Peter/Elfi/Dagmar/Heinz/Gerlinde/']
•ChaOrd(Name,OrdName,'/')
•Dir(c:\windows\*.*,dir)
•ChaOrd(dir{1})
```

Alias: (ChaArg2) Befehl: ChaOrdCha

Ordnen einer Liste durch eine weitere Liste

SYNTAX •**ChaOrdCha(*ListennameC1,ListennameC2,ListennameC3,Trennz*)**

BESCHREIBUNG Schreibt die Elemente der Liste C1 in die Liste C3 an den Positionen, die in der Liste C2 definiert sind

ListennameC1	:	Name der Variablen Liste C1
ListennameC2	:	Name der Variablen Liste C2
ListennameC3	:	Name der Variablen Liste C3
Trennz	:	Trennzeichen zwischen den Worten immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Die Liste ***C2*** muß immer ein Leerzeichen als Trennzeichen haben. Wenn die Liste ***C3*** nicht existiert, wird sie erzeugt.

BEISPIEL

```
[Unordnung:=´Autoren /System /Windows /Das /EGO /´]
[Tabelle:=´2 3 5 1 4 ´]
•ChaOrdCha(Unordnung,Tabelle,Ordnung,´/´)
[Ordnung]
••Die Anzeige des Inhalts der Variablen Liste Ordnung
••bewirkt die Ausgabe: Das/Autoren/System/EGO/Windows/
```

Alias: (ChaArg) Befehl: ChaOrdTab

Ordnen einer Liste durch eine Tabelle

SYNTAX •**ChaOrdTab(*ListennameC1,Tabellenname,ListennameC3,Trennz*)**

BESCHREIBUNG Speichert die Elemente der Liste C1 in der Liste C3 unter den Positionen ab, die in der Tabelle Tabellenname definiert wurden.

ListennameC1	:	Name der Variablen Liste C1
Tabellenname	:	Name der Tabelle
ListennameC3	:	Name der Variablen Liste C3
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wird kein Trennzeichen angegeben, erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Wenn die Liste ***C3*** nicht existiert, wird sie angelegt.

BEISPIEL

```
[Unordnung:=´Autoren /System /Windows /Das /EGO /´]
[Tabelle:=´2 3 5 1 4 ´]
•TabDef(Tabelle,5)
•Sam(Tabelle{1})•ChaSam(Klasse)
•ChaOrdTab(Unordnung,Tabelle,Ordnung,´/´)
[Ordnung]
••Die Anzeige des Inhalts der Variablen Liste Ordnung
••bewirkt die Ausgabe:
Das /Autoren /System /EGO /Windows /
```

Alias: (ChaSubst) Befehl: ChaPos

Schreiben eines Wortes an einer angegebener Position in einer Variablen Liste

SYNTAX •**ChaPos(*ListennameC1,Wort,Position,Trennz*)**

BESCHREIBUNG Schreibt bzw. ersetzt das Element ***Wort*** an die Position ***Position*** der Variablen Liste C1.

ListennameC1	:	Name der Variablen Liste C1
Wort	:	Einzutragendes Wort
Position	:	Position des einzutragenden Wortes
Trennz	:	Trennzeichen zwischen den Worten Immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf Basis des Leerzeichens.

BEISPIEL

```
[Folge:='x/x/x/x/x/x/']
•ChaPos(Folge,'Heinrich',4,'/')
[Folge]
••es erscheint:
x/x/x/Heinrich/x/x
```

Alias: (ChaRam) Befehl: ChaSam

Einsammeln der Zahlen einer Zeichenkette

SYNTAX •**ChaSam(*Listenname,AnzWort,Trennz)***

BESCHREIBUNG Sammelt die Zahlen ein, die im Inhalt einer alphanumerischen Variablen vorkommen

Listenname	:	Name der alphanumerischen Variablen
AnzWort	:	Index des Wortes in der alphanumerischen Variablen, in dem sich die Sammlung vollziehen wird
Trennz	:	Trennzeichen zwischen den Wörtern immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf Basis des Leerzeichens.

ANMERKUNG Zahleneinsammler müssen mit Hilfe des Befehls •**SamList**() vor dem Gebrauch des Befehls •**ChaSam**() definiert werden.

Wenn man nur den Namen der alphanumerischen Variablen angibt, wird deren gesamter Inhalt gesammelt.

BEISPIEL

```
•SamList(Rot,Grün,Blau,Weiß,Gelb)
[Wert:='255/123/128/10/32/']
•Sam(Rot)
•ChaSam(Wert)
```

SIEHE AUCH •**Sam**(), •**SamList**()

Alias: (ChaTab) Befehl: ChaTab

Kopieren eine Zeichenkettendatei in eine Zahlentabelle

SYNTAX •**ChaTab**(***Dateiname,Tabellenname***)

BESCHREIBUNG Kopiert Zahlen, die in einer Zeichenkette vorkommen, in eine Zahlentabelle

Dateiname : Name der Datei

Tabellenname : Name der Tabelle

ANMERKUNG Eine Zeichenkettendatei wird durch den Befehl •**TabCha**() erzeugt, der das Gegenteil von •**ChaTab**() bewirkt.

Die Zahlentabelle muß bereits definiert sein, bevor dieser Befehl benutzt wird. Wie die Tabelle dimensioniert ist und wieviele Elemente sie enthält, hat keine Bedeutung.

BEISPIEL

```
•TabDef(Tabelle,20,30)
•ChaTab(Speichertab,Tabelle)
```

SIEHE AUCH •**TabCha**()

Alias: (ChaTxt) Befehl: ChaTxt

Kopieren einer Zeichenkette in eine Text-Datei

SYNTAX •**ChaTxt(*ZKetteDateiname,TextDateiname*)**

BESCHREIBUNG Kopieren einer Zeichenkette ***ZKetteDateiname*** in eine ASCII- Textdatei ***TextDateiname***

ZKetteDateiname : Name der Zeichenkettendatei

TextDateiname : Name der Textdatei

ANMERKUNG Es ist keine Dateierweiterung *.cha nach dem Namen der Zeichenkettendatei anzugeben, aber es müssen alle Angaben der Dateierweiterung für den Namen der Textdatei angegeben werden.

Wenn die Textdatei erstellt ist, kann sie in allen Texteditoren verarbeitet und gedruckt werden.

BEISPIEL

```
•ChaTxt(Ergebnis,Ergebnis.txt)
```

Alias: (ChaOccur) Befehl: ChaVork

Ermitteln der Anzahl des Vorkommens eines Wortes in einer Liste

SYNTAX •**ChaVork(*ListennameC1,Wort,Position,Anzahl,Trennzeichen*)**

BESCHREIBUNG Ordnet der numerischen Variablen ***Anzahl*** die Anzahl des Vorkommens des Wortes ***Wort*** in der Liste C1 zu und erzeugt die Liste C2, die die Indizes dieser Vorkommen enthält.

ListennameC1	:	Name der Variablen Liste C1
Wort	:	Wort, das in C1 gesucht wird
Position	:	Diese Variable enthält die Nummern der Position(en), an denen das gesuchte Wort vorkommt
Anzahl	:	Diese numerische Variable enthält die Anzahl des Vorkommens des gesuchten Wortes
Trennzeichen	:	Trennzeichen zwischen den Worten Immer zwischen APOSTROPHEN. Wenn kein Trennzeichen angegeben wird, erfolgt die Operation auf Basis des Leerzeichens.

BEISPIEL

```
[Ergebnis:='schlecht/gut/sehr gut/gut/gut/']
•ChaVork(Ergebnis,'gut',PosGut,Anzahl,'/')
Positionen von GUT: [PosGut]    ••Positionen = 2/4/5
Vorkommen von GUT: [Anzahl]     ••Anzahl = 3
```

Alias: (Clip) Befehl: Clip

Aktivieren bzw. desaktivieren eines Bildschirmbereiches

SYNTAX •`Clip(+)` oder •`Clip(-)`

BESCHREIBUNG Dieser Befehl ermöglicht die Aktivierung oder Desaktivierung eines Bildschirmbereiches der durch den Befehl •`ClipDef`() definiert wurde.

•***`Clip(+)`***	:	aktiviert den Bildschirmbereich
•***`Clip(-)`***	:	desaktiviert den Bildschirmbereich

SIEHE AUCH •`ClipDef`()

Alias: (DefClip) Befehl: ClipDef

Definieren eines Bildschirmausschnittes (Clipping)

SYNTAX •**ClipDef(*Xhg,Yhg,Xbd,Ybd*)**

BESCHREIBUNG Bestimmt eine rechteckige Zone am Bildschirm, in der die Anzeigen aktiv sind.

Xhg,Yhg,Xbd,Ybd : Koordinaten der Diagonale des Bereiches

ANMERKUNG Wird nichts anderes angegeben, so gilt der gesamte Bildschirm als Ausgabezone. Sobald der Befehl ausgeführt wurde, ist die Zone aktiviert.

Um zu einer Ausgabe am Gesamtbildschirm zurückzukommen, ist der Befehl •**Clip**(-) zu verwenden.

BEISPIEL

```
•ClipDef(200,200,400,300)
•Objekt(Foto.PCX,0,0)
•Vektor(0,400,300,45)
```

SIEHE AUCH •**Clip**()

Alias: (CoTache) Befehl: CoAufg

Aufruf einer CoAufgabe

SYNTAX •**CoAufg(ModulName,p1,p2,...,pN)**

BESCHREIBUNG Laden und ausführen einer CoAufgabe

ModulName : Name des Moduls

p1, .. , pn : Zu übertragende Parameter an das Modul

ANMERKUNG Eine Co-Aufgabe ist ein Modul, das mit dem Befehl •**CoAufg()** aufgerufen wird, ausgehend von einer Aufgabe oder einer Co-Aufgabe, kehrt in die aufrufende Aufgabe, auf die dem Aufruf folgenden Zeile zurück, wenn es auf einen Befehl •Ende() trifft. Die aufgerufene CoAufgabe kann mit dem Befehl •**ModParam()** die vorherigen Parameter wiedererlangen. Es können bis zu 32 vorherige Parameter wiedererlangt werden. Eine Co-Aufgabe kann eine Co-Aufgabe oder eine Aufgabe aufrufen. Es ist möglich, bis zu 16 Niveaus von Co-Aufgaben zu verschachteln. Um eine Co-Aufgabe zu stoppen, ist es notwendig, den Namen der Ausgangs-Aufgabe mit dem Befehl •**AufgPause()** zu durchlaufen. Um die Ausführung einer Co-Aufgabe wieder aufzunehmen, ist es notwendig, den Namen der Ausgangs-Aufgabe mit dem Befehl •**AufgFort()** zu durchlaufen.

BEISPIEL

```
••Vor dem Aufruf der Co-Aufgabe
•CoAufg(CAufg)
••Nach dem Aufruf der Co-Aufgabe
```

SIEHE AUCH •**ModParam**(), •**Weiter**(), •**Ende**(), •**Aufgabe**(), •**CoMod**(), •**ModVerb**(), •**AufgPause**(), •**AufgFort**()

Alias: (CoMod) Befehl: CoMod

Aufrufen eines Co-Moduls

SYNTAX •**CoMod**(***Modulname,p1,p2, ...,pn***)

BESCHREIBUNG Lädt ein Modul und führt es aus.

Modulname : Name des Moduls

p1,p2,..pn : Parameter beim Übergang zum Modul

ANMERKUNG Ein Co-Modul ist ein Modul, das mit dem Befehl •**CoMod**() aufgerufen wird. Sobald es den Befehl •**Ende**() antrifft, kehrt es in das aufrufende Modul zurück, und zwar zur Zeile, die dem Aufruf folgt.

Das aufgerufene Co-Modul kann die Parameter wiedererlangen, die vom vorhergehenden Modul übernommen wurden. Dies geschieht mit dem Befehl •**ModParam**(). Es können bis zu 32 Parameter übernommen werden.

Ein Co-Modul kann seinerseits Co-Module oder Module aufrufen. Es ist möglich, bis zu 16 Niveaus von Co-Modulen zu generieren.

BEISPIEL

```
••Vor dem Aufruf des Co-Moduls
•CoMod(CMod1)
••Nach dem Aufruf des Co-Moduls
```

SIEHE AUCH •**Weiter**(), •**Ende**(), •**Aufgabe**(), •**ModVerb**(), •**ModParam**()

Alias: (FiCop) Befehl: DateiKopie

Kopieren einer Datei

SYNTAX •**DateiKopie(*Quelle,Ziel*)**

BESCHREIBUNG Kopiert eine oder mehrere Dateien.

Quelle	:	Name der Quelldatei oder Gruppe von Quelldateien
Ziel	:	Name der Zieldatei oder Gruppe von Zieldateien

BEISPIEL

```
•DateiKopie(D:\Folge\*.cha,D:\')
•DateiKopie(C:\Motor\Schüler.cha,A:\Notiz.cha)
```

SIEHE AUCH •**DateiUmben**(), •**DateiLösch**()

Alias: (FiSup) Befehl: DateiLösch

Löschen einer Datei

SYNTAX •**DateiLösch(*Dateiname*)**

BESCHREIBUNG Löschen einer Datei.

Dateiname : Name der Datei

BEISPIEL

```
•DateiLösch(D:\Folge\Folge.cha)
```

SIEHE AUCH •**DateiKopie**(), •**DateiUmben**()

Alias: (OuvreFichier) Befehl: DateiÖffnen

Aktivieren eines Dialogfensters zum Öffnen von Dateien

SYNTAX •**DateiÖffnen(*Auswahl,Titel,Pfad,Filter*)**

BESCHREIBUNG Ermöglicht das Öffnen eines Standard-Windows Dialogfensters zum Öffnen von Dateien.

Auswahl	:	Name der Zeichenketten-Variablen, die den Namen der ausgewählten Datei enthält
Titel	:	Text, der als Fenster-Titel erscheint (Standardeinstellung -Öffnen-)
Pfad	:	Name des aufgelisteten Verzeichnisses bei Öffnung des Dialogfensters
Filter	:	Filter für die Suche der Dateien im ausgewählten Verzeichnis

ANMERKUNG Die Filter haben eine besondere Form, deren Syntax wie folgt aussieht:

Text für die Suche1 /*.XXX;*.YYY;*.ZZZ / Text für die Suche2 /*.AAA;*.BBB/....

BEISPIEL

```
•DateiÖffnen(Auswahl,,'C:\Kurs',
                    EGO-Dateien/*.EGO;*.CHA/Grafiken/*.BMP)
```

SIEHE AUCH •**DateiSichern**()

Alias: (SauveFichier) Befehl: DateiSichern

Aktivieren eines Dialogfensters für das Sichern von Dateien

SYNTAX •**DateiSichern(*Auswahl,Titel,Pfad,Filter*)**

BESCHREIBUNG Ermöglicht das Öffnen eines Standard-Windows Dialogfensters für das Sichern von Dateien.

Auswahl	:	Name der Zeichenketten-Variablen, die den Namen der ausgewählten Datei enthält
Titel	:	Text, der als Fenster-Titel erscheint (Standardeinstellung - Sichern unter)
Pfad	:	Name des Verzeichnisses und der vorgeschlagenen Datei
Filter	:	Filter für die Suche der Dateien im ausgewählten Verzeichnis

ANMERKUNG Die Filter haben eine besondere Form, deren Syntax wie folgt aussieht:

Text für die Suche1/*.XXX;*.YYY;*.ZZZ / Text für die Suche2 /*.AAA;*.BBB/....

BEISPIEL

```
•DateiSichern(Auswahl,,'C:\Math\Berech.EGO',
                  EGO-Dateien/*.EGO;*.CHA/Grafiken/*.BMP)
```

SIEHE AUCH •**DateiÖffnen**()

Alias: (FiRen) Befehl: DateiUmben

Umbenennen einer Datei

SYNTAX •**DateiUmben(*NameAlt,NameNeu*)**

BESCHREIBUNG Ändert den Namen einer Datei.

NameAlt	:	Name der Datei
NameNeu	:	Neuer Name der Datei

BEISPIEL

```
•DateiUmben(D:\Folge\Folge.cha,Notiz.cha)
```

SIEHE AUCH •**DateiKopie**(), •**DateiLösch**()

Alias: (Dat) Befehl: Datum

Ermitteln des Systemdatums

SYNTAX •**Datum(*VarJahr,VarMonat,VarTag,Wochentag*)**

BESCHREIBUNG Ermittelt das Systemdatum und speichert es in die numerischen Variablen VarJahr, VarMonat, VarTag ab

VarJahr	:	Name der Variablen, in der das Jahr gespeichert wird
VarMonat	:	Name der Variablen, in der der Monat gespeichert wird
VarTag	:	Name der Variablen, in der der Tag gespeichert wird
Wochentag	:	Name der Variablen, in der der Wochentag gespeichert wird

ANMERKUNG Wenn man an Stelle der numerischen Variable Zahlen angibt, so wird das Systemdatum anhand dieser Zahlen aktualisiert.

BEISPIEL

```
•Datum(Jahr,Monat,Tag)
Heute ist der [Tag]:[Monat]:[Jahr]
```

SIEHE AUCH •**Uhrzeit**()

Alias: (DDEcmd) Befehl: DDEBef

Senden von DDE-Befehlen

SYNTAX •**DDEBef(*Servername,Thema,Detail,Befehl*)**

BESCHREIBUNG Schickt Mitteilungen oder Befehle an einen DDE Server

Servername	:	Name des DDE Servers
Thema	:	Gegenstand des DDE-Informationsaustausches
Artikel	:	Präzises Element des DDE-Informationsaustausches
Befehl	:	Befehl, den der Server ausführen soll

ANMERKUNG Üblicherweise werden die DDE Befehle zwischen Klammern geschrieben **[ ]** ; in diesem Fall muß man den Befehl zwischen Apostrophen schreiben oder die eckigen Klammern durch geschwungene Klammern **{ }** ersetzen.

BEISPIEL

```
•DDEBef(TrunTime,ANIMATION,,´[START(GPS)]´)
oder
•DDEBef(TrunTime,ANIMATION,,{START(GPS)})
```

SIEHE AUCH •**DDENachr**(), •**DDESetze**(), •**DDEWert**(), •**Extern**()

Alias: (DDEavis) Befehl: DDENachr

Aufheben des Ruhezustandes von EGO durch ein Wecksignal eines DDE-Servers

SYNTAX •**DDENachr(*Servername,Thema,Detail,VariableWert*)**

BESCHREIBUNG Dieser Befehl unterbricht den Ruhezustand von EGO mit Hilfe eines DDE-Servers, wenn dieser einen Wert in Zusammenhang mit Thema und Detail übersendet.

Servername	:	Name des DDE Servers
Thema	:	Gegenstand des DDE-Informationsaustausches
Detail	:	Präzises Element des DDE-Informationsaustausches
VariableWert	:	Name der Variablen, die den übersandten Wert durch den DDE-Server enthält

ANMERKUNG Wenn dieser Befehl ausgeführt ist, wartet EGO ausschließlich auf eine Weck-Nachricht von Seiten des Servers. Das heißt, daß die sequentielle Ausführung der Befehle im Modul angehalten wird und nur mit Hilfe der Nachricht des Servers wieder aufgenommen wird.

BEISPIEL

```
••DDEpo2  berechnet alle 6 Sekunden die Einwohnerzahl
••der USA
•DDENachr(DDEpo2,'US_Population','USA',var)
Die US-Einwohnerzahl beträgt zur Zeit [var]
•Schleife(=-2)
```

SIEHE AUCH •**DDEWert**(), •**DDESetze**(), •**Extern**()

Alias: (DDEposte) Befehl: DDESetze

Setzen eines Wertes in einem DDE-Server

SYNTAX •**DDESetze(*Servername,Thema,Variablenname,Wert*)**

BESCHREIBUNG Ordnet der Variablen ***Variablenname*** des DDE-Servers Servername einen Wert ***Wert*** zu

Servername	:	Name des DDE Servers
Thema	:	Gegenstand des DDE-Informationsaustausches
Variablenname	:	Name der DDE-Variablen des Servers, die den neuen Wert empfängt
Wert	:	Wert, der der DDE-Variablen zugeordnet wird

BEISPIEL

```
•DDESetze(Truntime,ANIMATION,,GESCHWINDIGKEIT,100)
```

SIEHE AUCH •**DDEBef**(), •**DDENachr**(), •**DDEWert**(),

Alias: (DDEvaleur) Befehl: DDEWert

Übernehmen eines Wertes von einem DDE-Server

SYNTAX •**DDEWert(*Servername,Thema,Variablenname,VariableWert*)**

BESCHREIBUNG Übernimmt einen Wert in der Variablen-Kette ***VariableWert***, die der Variablen ***Variablenname*** des DDE Servers Servername entspricht.

Servername	:	Name des DDE Servers
Thema	:	Gegenstand des DDE-Informationsaustausches
Variablenname	:	Name der DDE-Variablen des Servers, die den Wert enthält
VariableWert	:	Name der Variablen Kette, die den Wert empfängt, der vom DDE Server übernommen wird

BEISPIEL

```
•DDEWert(Progman,Progman,´Hauptgruppe´,VarGruppe)
```

SIEHE AUCH •**DDEBef**(), •**DDENachr**(), •**DDESetze**()

Alias: (Dir) Befehl: Dir

Auflisten des Inhaltes eines Verzeichnisses

SYNTAX •**Dir(*Suche,Dateiname,+*)**

BESCHREIBUNG Schreibt in die Zeichenkettendatei ***Dateiname*** die Liste der Verzeichnisse, die den Kriterien, die unter Suche angegeben wurden, entsprechen.

Suche	:	Datei-Suche, wie mit dem DOS Befehl Dir
Dateiname	:	Name der Datei, in welcher das Resultat abgespeichert wird

ANMERKUNG Es ist möglich, eine ***Suche*** nach verschiedenen Datei-Formaten durchzuführen. Der Parameter ***Suche*** beinhaltet die verschiedenen gesuchten Dateiformate, getrennt durch einen Strichpunkt (;). Um die Liste der Verzeichnisse eines Pfades zu erhalten, muß man für die Suche angegeben: ***Pfad*.*** .. Um im gesamten Baum eines Verzeichnises zu suchen, muß man als Wert für die ***Suche Pfad*./S*** angeben (nur bei Verzeichnissen anzuwenden). Wenn der Parameter **+** gesetzt ist, dann ermittelt der Befehl die Liste der Verzeichnisse mit der Größe und deren Erstellungsdatum ((nur bei Verzeichnissen anzuwenden).

BEISPIEL

```
•Dir(D:\EGO\*.BMP,BILDER)
•Dir(D:\EGO\*.BMP;D:\EGO\*.GIF;D:\BILD\BI*.PCX,BILDER)
•Dir(C:\*.,VERZ_C)
•Dir(C:\*./S,VERZ_C)
```

Alias: (DLL) Befehl: DLL

Aufrufen einer DLL-Funktion (Dynamic-Link Library)

SYNTAX •**DLL(*VarNameDLL,FuName,ErgVariable,Param1,..ParamN*)**

BESCHREIBUNG Ruft eine DLL-Funktion auf, die vorher mit dem Befehl •**DLLLaden**() geladen sein muß.

VarNameDLL	:	Name der Identifikationsvariablen der DLL
FuName	:	Name der auszuführenden Funktion
ErgVariable	:	Name der Variablen, in der der Wert des Ergebnisses der Funktion gespeichert wird
Param1...N	:	Parameter der Funktion (max. 20)

ANMERKUNG Die Namen der Variablen und Parameter müssen typisiert sein. Die Syntax und die Typisierung sind folgende:

!c =	Typ Zeichenkette	1024 Zeichen max
!L =	Typ integer long	32 bit
!n =	Typ integer	16 bit
!pr=	Typ pointer auf real	Real IEEE 64 bit
!tn=	Typ pointer auf array von integer	
!tL=	Typ pointer auf array von integer long	
!tr =	Typ pointer auf array von reaL	

Um einen Typ pointer zu definieren, ist es notwendig, den Buchstaben p dem Buchstaben des Typs voran zusetzen (ausgenommen für die Typ **tn**, **tL** und **tr**). Für einen Pointer Zeichenkette muß man z.B. schreiben: **!pc**. Für den Typ real ist nur der Typ pointer auf real (**!pr**) zulässig.

ACHTUNG: Für ***ErgVariable*** wird nur **!n**, **!L** und **!c** akzeptiert!

Vergessen Sie nicht, DLL nach deren Verwendung mit dem Befehl •**DLLFrei**() zu schließen.

BEISPIEL

```
•• DLL KERNEL von Windows
•DLLLaden(Kernel,KernelDll) ••Lädt die DLL Kernel
•DLL(KernelDLl,getfreespace, mem!l, 0!n)
freier Speicher = [mem / 1024 / 1024] MB
•DLLFrei(KernelDll) ••schließe DLL Kernel
```

SIEHE AUCH •**Extern**(), •**DLLFrei**(), die Funktion SYS() des Rechners

Alias: (LibDLL) Befehl: DLLFrei

Schließen einer DLL (Dynamic-Link Library)

SYNTAX •**DLLFrei(*VarIdent*)**

BESCHREIBUNG Schließt eine DLL, die zuvor mit dem Befehl •**DLLLaden**() geladen wurde.

VarIdent : Name der zu befreienden Identifikations-Variablen der DLL

BEISPIEL

```
••ruft eigenständig eine Funktion unter Windows auf
•DLLLaden(KernelDLL,krnl)
•DLL(krnl,SpeicherFreimachen,mem!1,0,!n)
Freier Speicher = [mem/1024] kB
•DLLFrei(krnl)
```

SIEHE AUCH •**DLL**(), •**DLLLaden**()

Alias: (ExecDLL) Befehl: DLLLaden

Laden einer DLL (Dynamic-Link Library)

SYNTAX •**DLLLaden(*DLLName,VarIdent*)**

BESCHREIBUNG Lädt ein DLL.

DLLName	:	Name der DLL, eventuell mit vollständigem Pfad
VarIdent	:	Name der Variablen, in der der Identifikationswert der DLL gespeichert wird

ANMERKUNG Die Identifikationsvariable ***VarIdent*** wird benutzt, sobald eine DLL Funktion mit dem Befehl •**DLL**() aufgerufen oder sobald die DLL mit dem Befehl •**DLLFrei**() geschlossen wird.

BEISPIEL

```
•• Laden der Windows DLL Kernel
•DLLLaden(Kernel,KernelDLL)
```

SIEHE AUCH •**DLLFrei**(), •**DLL**()

Alias: (Mul) Befehl: DoppelWorte

Ermitteln der Doppelworte in der Antwort des Schülers

SYNTAX

- **`DoppelWorte(+)`** oder
- **`DoppelWorte(+,W1,W2,...,Wn)`** oder
- **`DoppelWorte(-)`**

BESCHREIBUNG Erlaubt anläßlich der Analyse im streng konformen Modus die mehrfache Verwendung eines Wortes in der Antwort des Schülers. Die Systemvariablen ***ewNbMots*** und ***ewNbReconnus*** werden daher korrekt zugeordnet.

- ***`DoppelWorte(+)`*** : Erlaubt die mehrfache Verwendung ALLER Worte
- ***`DoppelWorte(+,W1,..,Wn)`*** : Erlaubt die mehrfache Verwendung der Wörter ***W1..Wn***
- ***`DoppelWorte(-)`*** : Desaktiviert den Befehl

BEISPIEL

- `DoppelWorte(+,Geschwindigkeit,Kraft,Anstrengung)`

Alias: (ImpCmd) Befehl: Druck

Definiert und erstellt eine Druckseite

SYNTAX

•**Druck**(***?,Aufl,Richtung***)

•**Druck**(***DokName,Aufl,Richtung,KopieZl***)

•**Druck**(**-**)

BESCHREIBUNG Definiert und erstellt eine Druckseite

•***Druck(?,Aufl,Richtung)***

: Ermittelt die Auflösung und die Druckrichtung der Seite

•***Druck(DokName,Aufl,Richtung,KopieAnz)***

: Definiert und erstellt ein Druckseite

•***Druck(-)*** : Schließt und sendet die Seite an den Drucker

? : Festgelegtes Zeichen für das Ermitteln der Auflösung

DokName : Name der dem Dokument gegeben wird

Aufl : Gewünschte Auflösung (ohne Angabe ist die Auflösung angepaßt an die Funktion des Druckers)

Richtung : Druckrichtung des Papiers
0 = Hochformat
1 = Querformat
Ohne Angabe ist das Hochformat eingestellt

KopieAnz : Legt die Anzahl der Kopien fest
(ohne Angabe = 1 Kopie)

ANMERKUNG Ohne Angabe ist die Auflösung der zu druckenden Seite durch die Funktion des angeschlossenen Druckers bestimmt. Um diese Auflösung zu kennen, kann man vorher den Befehl •**Druck*(?)*** ausführen lassen, welcher die Anzahl der horizontalen und vertikalen Punkte in die Systemvariablen ***ewPrnX*** und ***ewPrnY*** speichert. Um einen Druck zu beginnen, muß zuerst die Druckseite mit den Befehl •**Druck(DokName,Aufl,Richtung,KopieAnz)** geöffnet werden. Danach werden alle Befehle zur Darstellung auf diese Druckseite bezogen. Wenn die Seite fertiggestellt ist, muß man den Befehl •**Druck*(-)*** ausführen, um die Druckseite zu schließen und an den Drucker zu senden.

Es ist möglich, die Auflösung mit Hilfe des Parameters ***Aufl*** anläßlich der Eröffnung einer Druckseite zu ändern. Dieser Parameter legt die horizontalen Punkte der Seite fest, die vertikale Anzahl der Punkte wird in Zusammenspiel von Seitenformat und Druckrichtung berechnet.

Alle Befehle zur Darstellung können in einer Druckseite verwendet werden.

BEISPIEL

```
•Druck(Demo)
•KomFen(10,100,100,100,236,255)•Ausrichtung(3)
Bla Bla Bla ...
Bla Bla Bla ...
Bla Bla Bla ...
•Bild(Photo.pcx,,200,200,100,100,,2)
•Druck(-)
```

SIEHE AUCH •**BschDruck**()

Alias: **(Inter)** Befehl: **Eingabe**

Eine Eingabe des Schülers ermöglichen (Interventionspunkt)

SYNTAX •**Eingabe()**

BESCHREIBUNG Hält die Ausführung des Moduls an und wartet auf eine Reaktion des Schülers (Tastendruck oder Mausklick)

ANMERKUNG Dieser Befehl kann jederzeit im Textteil oder im Kommentarteil verwendet werden.

Er kann mit dem Befehl zur Öffnung eines Antwortfensters •**AntwFen**() gekoppelt werden.

BEISPIEL

```
•AntwFen(100,300,300,40,10,12)
•Eingabe()
```

SIEHE AUCH •**EingabeErl**(), •**AntwFen**()

Alias: (Pie) Befehl: EingabeErl

Verwaltung des Interventionspunktes des Schülers

SYNTAX •**EingabeErl(+)** oder •**EingabeErl(-)**

BESCHREIBUNG Dieser Befehl ermöglicht oder entzieht dem Schüler die Eingabemöglichkeit. Im letzteren Fall geht das Programm direkt vom Textteil zum Analyseteil über, ohne eine Eingabe des Schülers abzuwarten.

- •***EingabeErl(+)*** : ermöglicht dem Schüler eine Eingabe
- •***EingabeErl(-)*** : entzieht dem Schüler die Eingabemöglichkeit

SIEHE AUCH •**SprungBlock**(), •**Eingabe**(), •**AntwFen**()

Alias: (Ellipse) Befehl: Ellipse

Zeichnen einer Ellipse

SYNTAX •**Ellipse(*Xc,Yc,Rx,Ry*)**

BESCHREIBUNG Zeichnet eine Ellipse mit dem Mittelpunkt ***Xc,Yc*** und den Radien ***Rx,Ry***

Xc,Yc	:	Koordinaten des Mittelpunktes der Ellipse
Rx,Ry	:	Große und kleine Achsen der Ellipse

ANMERKUNG Die Zeichnung erfolgt in der Farbe, dem Modus und dem Stil des Pinsels, der gerade aktuell ist.

BEISPIEL

```
•Ellipse(320,200,100,50)
```

SIEHE AUCH •**EllipseVoll**(), •**Kreis**(), •**Bogen**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Disque) Befehl: EllipseVoll

Zeichnen einer ausgefüllten Ellipse

SYNTAX •**EllipseVoll(*Xc,Yc,Rx,Ry*)**

BESCHREIBUNG Zeichnet eine ausgefüllte Ellipse mit dem Mittelpunkt ***Xc,Yc*** und den Radien ***Rx,Ry***.

Xc,Yc	:	Koordinaten des Mittelpunktes der Ellipse
Rx,Ry	:	Große und kleine Achse der Ellipse

ANMERKUNG Die Kontur der Ellipse wird mit der Farbe, dem Modus und den Stil des Pinsels gezeichnet, der gerade aktuell ist.

Ausgefüllt wird die Fläche mit der aktuellen Füllfarbe.

BEISPIEL

```
•EllipseVoll(320,200,100,50)
```

SIEHE AUCH •**Kreis**(), •**Ellipse**(), •**Bogen**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Fin) Befehl: Ende

Beenden der Ausführung eines Moduls

SYNTAX **•Ende()** oder **•Ende(Prozess,ModNr)**

BESCHREIBUNG Beendet die Ausführung eines Moduls oder eines Kurses. Beendet die Ausführung des Moduls und kehrt im Falle eines Co-Moduls, einerCo-Aufgabe oder einer Aufgabe in das aufrufende Modul zurück. Es ist auch möglich, direkt in ein Co-Modul oder in eine Co-Aufgabe zu gelangen, indem man die parameter ***Prozess*** und ***ModNr*** durchläuft.

ANMERKUNG Um direkt in ein Co-Modul oder in eine Co-Aufgabe zu gelangen (oder allgemeiner in ein Modul eines Prozesses) ohne alle Co-Module oder Co-Aufgaben eines Prozesses durchlöaufen zu müssen, ist es notwendig die Nummer des Prozesses und die Nummer plus 1 des Co-Moduls oder der Co-Aufgabe zu durchlaufen, auf die man zurückkommen möchte

BEISPIEL

```
•Ende()
•Ende(0,2) •• übergibt die Steuerung dem Modul 1 des Prozesses 0
```

SIEHE AUCH •**Weiter**()

Alias: (Exec) Befehl: Extern

Ausführen eines externen Programms

SYNTAX •**Extern(*Prog,Modus,X,Y,Schutz*)**

BESCHREIBUNG Lädt und führt ein externes Programm in einem Fenster aus, das durch Modus definiert wird.

Prog	:	Name des Programms, eventuell mit voran-stehendem Pfad und wenn nötig mit nachstehenden Parametern
Modus	:	Anzeigemodus des Ausführungsfensters 0,1,2,10,11,12
X,Y	:	Anzeigekoordinaten der oberen linken Ecke des Ausführungsfensters des Programms
Schutz	:	Indikator der Beeinflußbarkeit des externen Programms

Modus		**Fenster**
0	=	reduziert, Hintergrund
1	=	normal, Hintergrund
2	=	Vollbildschirm, Hintergrund
10	=	reduziert, immer im Vordergrund
11	=	normal, immer im Vordergrund
12	=	Vollbildschirm, immer im Vordergrund

Schutz		**Zustand**
0	=	Der Schüler kann das Programmfenster ver-schieben und/oder das Programm beenden
1	=	Der Schüler kann das Programmfenster weder verschieben, noch das Programm beenden

ANMERKUNG Im Modus ***0*** bis ***2*** wird das Fenster als im Hintergrund befindlich bezeichnet, da es hinter dem Fenster EGO abläuft, wenn der Schüler dieses anklickt oder wenn das Modul auf eine Aktion des Schülers wartet. Im Modus ***10*** bis ***12*** bleibt das Fenster immer vor dem Fenster EGO.

BEISPIEL

```
•Extern(Calc.exe,11,100,100,1)
```

SIEHE AUCH •**VBschLösch**(), •**DLLLaden**()

Alias: (SupExec) Befehl: ExternEnde

Beenden eines externen Programms

SYNTAX •**ExternEnde(*Programmname*)**

BESCHREIBUNG Beendet die Ausführung eines externen Programms, das mit dem Befehl •**Extern**() gestartet wurde.

Programmname : Name des externen Programms

BEISPIEL

```
•ExternEnde(Calc)
```

SIEHE AUCH •**Extern**()

Alias: (LibCoul) Befehl: FarbeFrei

Definieren einer verfügbaren Farbreihe für Animationen

SYNTAX •**FarbeFrei(*Anfangsindex,Endindex*)**

BESCHREIBUNG Freie Farben der aktuellen Palette. Dieser Befehl ermöglicht einer ***FLC*** Animation oder einem ***AVI***-Video, sich an die aktuelle Palette anzupassen.

Anfangsindex : Index der ersten frei verfügbaren Farbe

Endindex : Index der letzten frei verfügbaren Farbe

BEISPIEL

```
•FarbeFrei(10,120)
```

SIEHE AUCH •**FarbenRes**(), •**AAPLAY**()

Alias: (ResCoul) Befehl: FarbenRes

Festlegen der Farben im reservierten Modus

SYNTAX •**FarbenRes(*Anfangsindex,Endindex*)**

BESCHREIBUNG Reserviert die Farben der aktuellen Palette. Dieser Befehl ist das Gegenteil des Befehls •**FarbeFrei**() und ist die Standardeinstellung. Die Farben, auf die sich der Befehl •**RGBWechsel**() auswirkt, müssen reserviert werden.

Anfangsindex : Index der ersten Farbe, die reserviert werden soll

Endindex : Index der letzten Farbe, die reserviert werden soll

BEISPIEL

```
•FarbenRes(10,120)
```

SIEHE AUCH •**FarbeFrei**(), •**RGBWechsel**()

Alias: (CoulPB) Befehl: FarbePF

Ermitteln der Füllfarbe und Pinselfarbe

SYNTAX •**FarbePF(*Pinsel,Füllfarbe*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, die aktuellen Farben des Pinsels und der Füllfarbe zu ermitteln.

Pinsel	:	Variablenname, der die Farbe des Pinsels aufnimmt
Füllfarbe	:	Variablenname, der die Füllfarbe aufnimmt

BEISPIEL

```
•FarbePF(PAktuell,FAktuell)
```

SIEHE AUCH •**Pinsel**(), •**Füllfarbe**()

Alias: (Degrade) Befehl: FarbSkala

Erzeugen einer Farbskala

SYNTAX •**FarbSkala(*Anfangsindex,Endindex,Pal*)**

BESCHREIBUNG Erzeugt in der aktuellen Farbpalette Farbabstufungen zwischen den Farben mit den Indizes ***Anfangsindex*** und ***Endindex***, beide inklusive.

Anfangsindex	:	Index der ersten Farbe der Farbabstufung
Endindex	:	Index der letzten Farbe der Farbabstufung
Pal	:	Nummer der gewählten Palette (0 - 16)

ANMERKUNG Ohne Angabe ist die ausgewählte Palette die physische Palette (0).

BEISPIEL

```
•RGBWechsel(20,100,100,0)
•RGBWechsel(50,255,255,0)
•FarbSkala(20,50)
```

SIEHE AUCH •**RGBWechsel**(), •**RGBIndex**()

Alias: (Fen) Befehl: Fen

Aktivieren eines Windows-Fensters

SYNTAX •**Fen**(***FenNum,Auswahl,Sicht***)

BESCHREIBUNG Anzeigen und aktivieren eines Windows-Fensters.

FenNum	:	Nummer des Fensters
Auswahl	:	Parameter der Auswahl 0 = inaktiv, 1 = aktiv
Sicht	:	Parameter für ein verstecktes oder ein sichtbares Fenster 0 = versteckt, 1 = sichtbar (Standardeinstellung)

ANMERKUNG Der Befehl setzt ein Fenster aktiv, das mit dem Befehl •**FenDef**() definiert wurde und es wird angezeigt, wenn ein sichtbares Fenster festgelegt wurde (wenn Sicht = 1). Wenn ein Fenster ausgewählt ist (Auswahl = 1), dann erscheint es über den anderen vorhandenen Fenstern. Wenn ein Fenster aktiv gesetzt wurde und dieser Bildschirm 0 aktiv gesetzt ist, dann erscheinen alle Darstellungsbefehle in diesem Fenster.

BEISPIEL

```
•FenDef(2,HILFE-FENSTER,100,100,200,200,200,400)
•Fen(2,0)
•BschLösch(255)•Pinsel(236)•Font(arial,18)[I:=1]
[Hilfe{i}][i:=i+1]•Schleife(19,=)
```

SIEHE AUCH •**FenDef**(), •**FenFix**(), •**FenList**(), •**FenInfo**(), •**FenLösch**(), •**FenOrd**(), •**FenPos**(), •**KoordKonv**()

Alias: (DimFen) Befehl: FenDef

Definieren eines Windows-Fensters

SYNTAX •**FenDef(*FenNum,Titel,x,y,B,H,MaxB,MaxH,Rahmen,Attrib,Ini,Icon,Verb*)**

BESCHREIBUNG Definiert ein Windows-Fenster

FenNum	:	Nummer des zu erstellenden Fensters (1 bis 129)
Text	:	Text, der im Titel des Fensters erscheint
x,y	:	Koordinaten der linken oberen Ecke
B,H	:	Breite und Höhe des Fensters bei seiner Anfangsdarstellung
MaxB,MaxH	:	Maximale Breite und Höhe des Fensters
Rahmen	:	Art des Fenster-Rahmens
Attrib	:	Fenster-Attribute (Tasten zum Vergrößern / Verkleinern des Fensters)
Ini	:	Anfangszustand des Fensters (0 = reduziert, 1 = normal (Standardeinstellung), 2 = vergrößert)
Icon	:	Parameter zur Verwendung einer Icone für das Fenster (0 = kein Wechsel (Standardeinstellung), 1 = Neue Icone)
Verb	:	Nummer des Fensters, mit dem das Fenster ***FenNum*** verbunden ist

Rahmen		**Resultat**
0	=	Kein Rahmen
1	=	Schmaler Rahmen (Standardeinstellung)
2	=	Veränderbarer Rahmen
3	=	Breiter Rahmen

Attrib		**Attribute**
0	=	Ohne Attribute (Standardeinstellung)
1	=	Taste Menüsystem
2	=	Taste Verkleinerung
3	=	Attribute 1 und 2
4	=	Taste Vergrößerung
5	=	Attribute 1 und 4
6	=	Attribute 2 und 4
7	=	Attribute 1, 2 und 4**Fehler! Verweisquelle konnte nicht gefunden werden.**

ANMERKUNG

Einem Fenster können Nummern von 1 bis 129 zugewiesen werden, das Fenster 0 ist das Grundfenster des Moduls und beinhaltet den vollen Bildschirm ohne einen Titel, es kann mit Hilfe dieses Befehls auch umdimensioniert werden. Ein so erstelltes Fenster erscheint am Bildschirm, nachdem der Befehl •**Fen**() ausgeführt wurde. Um ein Fenster zu schließen, ist der Befehl •**FenLösch**() zu verwenden.

Der Parameter ***Icon*** , der auf 1 gesetzt ist, erlaubt das ersetzen der Icone, durch eine Icone, deren Bild 0,0 ist, also ohne aktiven Bildschirm (siehe auch •**MausCursB**() für das Format des Bildes). Im Gegensatz zum BildCursor kann er in Farbe sein.

Der Parameter ***Verb*** beinhaltet die Nummer des Fensters, mit dem das erstellte Fenster verbunden ist. Vebunden mit einem anderen Fenster bedeutet, daß bei versetzen eines Fensters alle verbundenen mitwandern. Im Fall einer Umdimensionierung des Fensters 0, und wenn man nicht den Hintergrundbildschirm am gesamten Bildschirm haben möchte, muß man die verschiedenen Befehle zur Fenstermanipulation durch den Befehl •**FenPos**() folgen lassen.

BEISPIEL

```
•DimFen(0,Fenster,0,100,100,400,400,400,400,1)
•FenFix(0,,,0,0)
•FenPos(0,100,100,400,400)
•Fen(0)
•FenDef(3,Demonstration,100,100,200,200,2,5)
•FenDef(25,,400,200,150,150)
•Fen(25,0)
•Fen(3,1)
```

SIEHE AUCH

•**Fen**(), •**FenFix**(), •**FenList**(), •**FenInfo**(), •**FenLösch**(), •**FenOrd**(), •**FenPos**(), •**KoordKonv**()

Alias: (FenFixe) Befehl: FenFix

Verriegeln/Entriegeln eines Windows-Fensters

SYNTAX •**FenFix(*FenNum,Versch,Aktiv,BLeisteH,BLeisteV*)**

BESCHREIBUNG Verriegeln / Entriegeln eines Windows-Fensters

FenNum	:	Nummer des Windows-Fensters
Versch	:	Parameter der festlegt, ob das Fenster verschiebbar ist oder nicht (0 = Fenster fix, 1 = Fenster verschiebbar)
Aktiv	:	Parameter zur Festlegung, ob ein Fenster aktivierbar ist oder nicht (0 = nicht aktivierbar, 1 = aktivierbar)
BLeisteH	:	Definition der Schritte für das Vorbeiziehen der horizontalen Bildlaufleiste 0 = ohne Bildlaufleiste, >0 = Veränderungswert (Increment) für das Verschieben 1 = Standardwert
BLeisteV	:	Definition der Schritte für das Vorbeiziehen der vertikalen Bildlaufleiste 0 = ohne Bildlaufleiste, >0 = Veränderungswert (Increment) für das Verschieben 1 = Standardwert

BEISPIEL

•FenFix(3,0,1)

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenList**(), •**FenInfo**(), •**FenLösch**(), •**FenOrd**(), •**FenPos**(), •**KoordKonv**()

Alias: (FenForme) Befehl: FenForm

Definieren einer nicht rechteckigen Kontur eines Windows-Fensters

SYNTAX •**FenForm(FenNum,VarTab)** oder •**FenForm(FenNum,Rx,Ry)**

BESCHREIBUNG Definieren einer nicht rechteckigen Kontur eines Windows-Fensters. Zwei verschiedene Konturen sind möglich: Kontur eines Vielecks oder eine Kontur einer Ellipse.

FenNum	:	Nummer des Windows-Fensters
VarTab	:	Name der Variablen-Tabelle, welche die Eck-Koordinaten der Vieleckkontur des Fensters beinhaltet
Rx,Ry	:	Zwei numerische Werte, welche die Radien der elliptischen Kontur des Fensters beschreiben.

ANMERKUNG Um einem Fenster seine normale rechteckige Kontur wiederzugeben, muß man den Befehl •**FenForm(FenNum)** ausführen.

BEISPIEL

```
•FenDef(3,,100,100,200,200,200,200,2)
•FenForm(3,100,150)
•Fen(3,1)
```

SIEHE AUCH •**FenDef**(), •**Fen**(), •**FenListe**(), •**FenPos**(), •**FenInfo**(), •**FenOrd**(), •**FenLösch**(), •**KoordKonv**()

Alias: (FenInfo) Befehl: FenInfo

Ermitteln der Objekt-Daten eines Windows-Fensters

SYNTAX •**FenInfo(*FenNum,Vx,Vy,VB,VH,Zust,VSx,VSy,Sicht,Verb*)**

BESCHREIBUNG Dieser Befehl erlaubt das Ermitteln der Objekt-Daten (Position, Breite, Höhe,...) eines Windows-Fensters.

FenNum	:	Nummer des Windows-Fensters
Vx,Vy	:	Name der numerischen Variablen, die die Koordinaten der linken oberen Ecke des Fensters enthält
VB,VH	:	Name der numerischen Variablen, die die Breite und die Höhe des Fensters enthält
Zust	:	Name der numerischen Variablen, die den Zustand des Fensters enthält (0 = verkleinert, 1 = normal, 2 = vergrößert)
VSx,VSy	:	Name der numerischen Variablen, die die Position der Cursorleiste der horizontalen oder vertikalen Bewegung im Fenster enthält
Sicht	:	Name der numerischen Variablen, die 0 enthält, wenn das Fenster unsichtbar ist und 1 wenn es sichtbar ist
Verb	:	Name der numerischen Variablen, die die Nummer des Fensters enthält, mit dem ***FenNum*** verbunden ist.

ANMERKUNG Die Koordinaten x und y, gespeichert in den Variablen ***Vx*** und ***Vy*** sind relativ an das Fenster gebunden, mit dem es verbunden ist. Wenn dieser Befehl in Zusammenhang mit einem nicht verbundenen Fenster verwendet wird, dann sind die Koordinaten relativ zum Fenster 0, so wie normal alle Fenster an dieses gebunden sind.

BEISPIEL

```
•FenInfo(1,F1X,F1Y)
•FenInfo(1,F1X,F1Y,F1B,F1H,,F1SX,F1SY)
•FenInfo(1,,,,,F1Zust)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenList**(), •**FenFix**(), •**FenLösch**(), •**FenOrd**(), •**FenPos**(), •**KoordKonv**()

Alias: (CopFen) Befehl: FenKopie

Kopie zwischen Fenstern

SYNTAX •**FenKopie(*FQ,FZ,Xq,Yq,Bq,Hq,Xz,Yz,Bz,Hz,Modus*)**

BESCHREIBUNG Erlaubt das Kopieren des gesamten Fensters oder eines Teils davon in ein beliebiges anderes Fenster

FQ	:	Quellfenster
FZ	:	Zielfenster
Xq,Yq	:	Koordinaten der linken hohen Ecke der Oberfläche des Quellfensters
bq,Hq	:	Breite und Höhe der Oberfläche des Quellfensters
Xz,Yz	:	Koordinaten der linken hohen Ecke des Zielfensters
bz,Hz	:	Breite und Höhe von der Oberfläche des Zielfensters, wohin die Oberfläche des Quellfenster kopiert werden soll.
Modus	:	Anzeigemodus (ohne Angabe: Kopie). Das ist der gleiche Modus wie für .•**Bild**()

ANMERKUNG Die Koordinaten ***Bz, Hz*** sind nur notwendig, wenn man die Größe des Zielfensters ändern will.

BEISPIEL

```
•FenKopie(1,3,100,100,150,170,100,270)
```

SIEHE AUCH •**BschKopie**(), •**Bild**()

Alias: (Fenlist) Befehl: FenList

Auflisten der Windows-Fenster

SYNTAX •**FenList(*VarName*)**

BESCHREIBUNG Dieser Befehl übernimmt in einer Liste die Nummern der vorhandenen Windows-Fenster

VarName : Name der Zeichenkette, die die Nummern der vorhandenen Windows-Fenster enthält

ANMERKUNG Die Nummern der vorhandenen Windows-Fenster werden in die Zeichenkette ***VarName*** mit dem Leerzeichen als Trennzeichen übernommen.

BEISPIEL

```
•FenList(FenListe)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenFix**(), •**FenInfo**(), •**FenLösch**(), •**FenPos**(), •**FenOrd**(), •**KoordKonv**()

Alias: (SupFen) Befehl: FenLösch

Löschen eines Windows-Fensters

SYNTAX •**FenLösch(*FenNum*)**

BESCHREIBUNG Löscht ein Windows-Fenster

FenNum : Nummer des zu löschenden Fensters

ANMERKUNG Wenn das Fenster, das man löscht, verbundene Fenster enthält, dann werden auch diese gelöscht. Wenn der Befehl ohne Parameter verwendet wird, werden alle Fenster geschlossen. Man kann auch eine Liste von zu löschenden Fenstern angeben, jede Nummer eines Fensters muß durch einen Beistrich getrennt werden.

BEISPIEL

```
•FenLösch(2)
•FenLösch(2,1,5,22)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenFix**(), •**FenInfo**(), •**FenList**(), •**FenOrd**(), •**FenPos**(), •**KoordKonv**()

Alias: (FenOrdre) Befehl: FenOrd

Definiert die Anordnung der Darstellung von Windows-Fenstern

SYNTAX •**FenOrd(*0,FenNr1,...FenNrN*)** oder •**FenOrd(*1,FenNr*)**

BESCHREIBUNG Dieser Befehl legt die Anordnung (Reihenfolge) von Windows-Fenstern fest oder es kann ein Fenster hinter alle anderen plaziert werden.

•***FenOrd(0, FenNr1,...FenNrN)***

: Stellt die Windows-Fenster in der Reihenfolge der Liste ***FenNr1, ... FenNrN*** dar

•***FenOrd(1,FenNr)*** : Plaziert das Fenster ***FenNr*** hinter alle anderen Fenster

BEISPIEL

```
•FenOrd(o,5,3,20,14,6)
•FenOrd(1,5)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenFix**(), •**FenList**(), •**FenInfo**(), •**FenLösch**(), •**FenPos**(), •**KoordKonv**()

Alias: (FenPos) Befehl: FenPos

Verändern und positionieren eines Windows-Fensters

SYNTAX •**FenPos(*FenNum,x,y,B,H,Zust,Sx,Sy,Sicht***

BESCHREIBUNG Dieser Befehl ermöglicht das Verändern des Aussehens und der Position eines Windows-Fensters.

FenNum	:	Nummer des Windows-Fensters
x,y	:	Koordinaten der linken oberen Ecke des Fensters
B,H	:	Breite und Höhe des Fensters
Zust	:	Name der numerischen Variablen, die den Zustand des Fensters enthält 0 = verkleinert, 1 = normal, 2 = vergrößert
Sx,Sy	:	Position der Cursorleiste für die horizontale bzw. vertikale Bildlaufleiste des Fensters
Sicht	:	Parameter für ein verstecktes oder ein sichtbares Fenster 0 = versteckt, 1 = sichtbar (Standardeinstellung)

ANMERKUNG Die Koordinaten **x** und **y**, sind relativ an das Fenster gebunden, mit dem es verbunden ist. Wenn dieser Befehl in Zusammenhang mit einem nicht verbundenen Fenster verwendet wird, dann sind die Koordinaten relativ zum Fenster 0, so wie normal alle Fenster an dieses gebunden sind

BEISPIEL

```
•FenPos(1,150,200)
•FenPos(1,200,200,100,100,,50,20)
•FenPos(1,,,,,0)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenFix**(), •**FenList**(), •**FenInfo**(), •**FenLösch**(), •**FenOrd**(), •**KoordKonv**()

Alias: (Boite) Befehl: Fläche

Zeichnen einer rechteckigen Fläche

SYNTAX •**Fläche(*Xh,Yh,Xb,Yb*)**

BESCHREIBUNG Zeichnet eine ausgefüllte rechteckige Fläche, die durch die Koordinaten der Diagonale definiert wird (***Xh,Yh,Xb,Yb***)

Xh,Yh,Xb,Yb : Koordinaten der Diagonale der Fläche

ANMERKUNG Die Darstellung erfolgt in der Farbe und dem Modus der aktuellen Füllfarbe.

BEISPIEL

```
•Füllfarbe(254)
•Fläche(100,150,400,200)
```

SIEHE AUCH •**Rechteck**(), •**RRechteck**(), •**Füllen**(), •**Füllfarbe**(), •**ZeiMod**()

Alias: (BoiteLH) Befehl: FlächeBH

Zeichnen einer rechteckigen Fläche

SYNTAX •**FlächeBH(*Xh,Yh,B,H*)**

BESCHREIBUNG Zeichnet eine ausgefüllte rechteckige Fläche, die durch die Koordinaten der linken oberen Ecke ***Xh,Yh*** und durch seine Breite ***B*** und seine Höhe ***H*** definiert wird.

Xh,Yh	:	Koordinaten der linken oberen Ecke der Fläche
B,H	:	Breite und Höhe der Fläche

ANMERKUNG Die Darstellung erfolgt in der Farbe und dem Modus der aktuellen Füllfarbe.

BEISPIEL

```
•Füllfarbe(254)
•FlächeBH(100,150,50,80)
```

SIEHE AUCH •**Fläche**(), •**Rechteck**(), •**RRechteck**(), •**Füllen**(), •**Füllfarbe**(), •**ZeiMod**()

Alias: (FlcFin) Befehl: FLCEnde

Schließen einer Animation die durch •**FLCLaden**() geladenen wurde

SYNTAX •**FLCEnde** (***AnimName***)

BESCHREIBUNG Schließen einer durch geladenenen Animation

AnimName : Dateiname der durch •**FLCLaden**() geladenen Animation

BEISPIEL

```
•FLCEnde(D:\ANIM\ANIM2.FLC)
```

SIEHE AUCH •**FLCInfo**(), •**FLCLaden**(), •**FLCListe**(), •**FLCPause**(), •**FLCSpiel**()

Alias: (FLCInfo) Befehl: FLCInfo

Übernehmen der Zustandsparameter einer FLC Sequenz

SYNTAX •**FLCInfo(*VarName,X,Y,B,H,Fen,Zustand,Seg,Pos*)**

BESCHREIBUNG Übernehmen der Zustandsparameter einer FLC Sequenz

VarName	:	Name einer Variable
X,Y	:	Name der Variablen, die die Koordinaten der linken oberen Ecke des Darstellungsfensters der FLC-Sequenz enthalten
B	:	Name der Variablen, die die Breite des Fensters enthält
H	:	Name der Variablen, die die Höhe des Fensters enthält
Fen	:	Name der Variablen, die die Nummer des Fensters enthält, die von der Sequenz verwendet wird
Zustand	:	Name der Variablen, die den Zustand der FLC-Sequenz enthält 0 = Beendet 1 = Pause 2 = Spielt 3 = Schleife
Seg	:	Name der Variablen, welche die Nummer des aktiven Elementes der FLC Sequenz enthält
Pos	:	Name der Variablen, welche die momentane Position in einer FLC Sequenz enthält. Die Einheit ist das Bild.

BEISPIEL

```
•FLCLaden(c:\flc\seq.FLC,,AniFLC)
•FLCSpiel(AniFLC,100,100,1)
•FLCInfo(AniFLC,,,,,,Zustand)
```

SIEHE AUCH •**FLCEnde**(), •**FLCLaden**(), •**FLCListe**(), •**FLCPause**(), •**FLCSpiel**()

Alias: (FlcCharge) Befehl: FLCLaden

Laden einer FLC-Animation

SYNTAX •**FLCLaden(*AnimName,Transp,Alias*)**

BESCHREIBUNG Laden einer FLC-Animation (die mit dem Programm Animator-Pro oder 3D Studio erstellt wurde)

AnimName	:	Dateiname der Animation mit Angabe des Pfades
Transp	:	Anzeigeoption der Animation mit Transparenz 0 = Keine Transparenz 1 = Mit Transparenz (Standardeinstellung)
Alias	:	Mit der FLC-Animation verknüpfter Alias-Name

ANMERKUNG Dieser Befehl lädt alle FLC-Animationen, die Transparenz-Option kann jedoch nur nach einer Compilierung der FLC-Datei durch eine externe Anwendung (Compiler EGI von ITB CompuPhase) erlangt werden.

BEISPIEL

```
•FLCLaden(D:\ANIM\ANIM1.FLC,0,ANIFLC)
•FLCSpiel(ANIFLC,100,100)
```

SIEHE AUCH •**FLCEnde**(), •**FLCInfo**(), •**FLCListe**(), •**FLCPause**(), •**FLCSpiel**()

Alias: (FLCList) Befehl: FLCListe

Ermittelt die Liste der Alias der geladenen FLC

SYNTAX •**FLCListe(*VarName,'Trennz'*)**

BESCHREIBUNG Ermittelt die Liste der geladenen FLC-Animationen.

VarName	:	Name der Zeichenketten-Variablen, die die Alias der geladenen FLC-Animationen enthält
Trennz	:	Trennzeichen für die Wörter in der Liste

ANMERKUNG Die Alias der geladenen FLC-Animationen sind in der Zeichenkette ***VarName*** enthalten, mit den Zeichen ***Trennz*** als Trennzeichen. Das Standard-Trennzeichen ist die vertikale Linie: |

BEISPIEL

```
•FLCListe(ListeFLC)
```

SIEHE AUCH •**FLCEnde**(), •**FLCInfo**(), •**FLCLaden**(), •**FLCPause**(), •**FLCSpiel**()

Alias: (FlcPause) Befehl: FLCPause

Unterbricht eine FLC-Animation (Pause-Funktion)

SYNTAX •**FLCPause(*AnimName*)**

BESCHREIBUNG Unterbricht eine FLC-Animation (die mit dem Programm Animator-Pro oder 3D-Studio erstellt wurde)

AnimName : Dateiname der Animation mit Angabe des Pfades

ANMERKUNG Dieser Befehl unterbricht alle FLC-Animationen, die durch den Befehl •**FLCSpiel**() aktiviert wurden. Um die Animation wieder zu starten verwendet man den Befehl •**FLCSpiel *(AnimName*)**.

BEISPIEL

```
•FLCLaden(D:\ANIM\ANIM1.FLC,0,AniFLC)
•FLCSpiel(AniFLC,150,50,1)
•Eingabe()
•FLCPause(AniFLC)
•Eingabe()
•FLCSpiel(AniFLC,,,1)
```

SIEHE AUCH •**FLCEnde**(), •**FLCInfo**(), •**FLCLaden**(), •**FLCListe**(), •**FLCSpiel**()

Alias: (FlcJoue) Befehl: FLCSpiel

Spielen einer FLC-Datei

SYNTAX •**FLCSpiel(*AnimName,X,Y,Wiederh,Geschw,Seg*)**

BESCHREIBUNG Spielen einer FLC-Animation (die mit dem Programm Animator-Pro oder 3D Studio erstellt wurde)

AnimName	:	Dateiname der Animation mit Angabe des Pfades
X,Y	:	Position der Animation
Wiederh	:	Parameter für Animations-Schleife 0 = Einmaliges Abspielen der Animation 1 = Abspielen der Animation in einer Schleife (Standardeinstellung)
Geschw	:	Geschwindigkeitsparameter. bestimmt durch den Zeitraum zwischen jedem Einzel-Bild, angegeben in Millisekunden 0 = Maximale Geschwindigkeit 70 = Normale Geschwindigkeit (Standardeinstellung)
Seg	:	Nummer des Spielsegmentes. Das erste Segment hat die Nummer 1

ANMERKUNG Dieser Befehl spielt alle FLC-Animationen, dagegen kann die Wiedergabe in Segmenten nur nach erfolgter Kompilierung der FLC-Datei durch eine externe Anwendung (Compiler EGI von ITB CompuPhase) erlangt werden.

BEISPIEL

```
•FLCLaden(D:\ANIM\ANIM1.FLC,0)
•FLCSpiel(D:\ANIM\ANIM1.FLC,150,50,1)
```

SIEHE AUCH •**FLCEnde**(), •**FLCLaden**(), •**FLCPause**()

Alias: (Police) Befehl: Font

Laden eines Fonts (Schrift) für den Text

SYNTAX •**Font(*Schrift,Größe,Stärke,Italic,Unterstr*)**

BESCHREIBUNG Lädt eine Schrift.

Schrift	:	Name der Schrift
Größe	:	Höhe der Buchstaben in Pixel
Stärke	:	Stärke der Buchstaben (0 bis 2) 0 = Fein 1 = Normal (Standardeinstellung) 2 = Fett
Italic	:	Italic-Attribute 0 = normal (Standardeinstellung) 1 = kursiv
Unterstr	:	Unterstreichung-Attribute 0 = unterstrichen (Standardeinstellung) 1 = normal

ANMERKUNG Die Standardeinstellung für die ***Stärke*** ist **1**, die für ***Italic*** ist **0** und für ***Unterstr*** ist **0.**

BEISPIEL

```
•Font(Arial,14,2,1)
```

SIEHE AUCH •**TasteFont**(), •**AntwFenFont**()

Alias: (InstPol) Befehl: FontInst

Installieren einer "TRUE TYPE" Schrift

SYNTAX •**FontInst(*Name*)**

BESCHREIBUNG Installiert unter Windows eine "TRUE TYPE" Schrift.

Name : Name der Schrift

ANMERKUNG Bevor man eine Schrift installiert, sollte man sich mit dem Befehl •**FontVorh**() vergewissern, daß diese Schrift noch nicht installiert ist. Das mehrfache installieren einer Schrift kann unangenehme Effekte hervorrufen.

BEISPIEL

```
•FontVorh('Brush Script Mt', Probe)
•Wenn([Probe=0])•FontInst(c:\windows\system\Brushci.ttf)
```

SIEHE AUCH •**FontLösch**(), •**FontVorh**()

Alias: (SupPol) Befehl: FontLösch

Deinstallieren einer "TRUE TYPE" Schrift

SYNTAX •**FontLösch(*Name*)**

BESCHREIBUNG Deinstalliert eine "TRUE TYPE" Schrift.

Name : Name der Schrift

BEISPIEL

```
•FontLösch(c:\windows\system\Arial.ttf)
```

SIEHE AUCH •**FontInst**(), •**FontVorh**()

Alias: (ExistePol) Befehl: FontVorh

Überprüft, ob eine bestimmte TRUE TYPE Schrift installiert ist

SYNTAX •**FontVorh(*FontName,VarName*)**

BESCHREIBUNG Überprüft, ob eine bestimmte TRUE TYPE-Schrift installiert ist.

FontName : Name der Schrift

VarName : Name der Variablen, die den ermittelten Wert enthält.

0 = Font ist nicht installiert

1 = Font ist installiert

ANMERKUNG Man muß den Anwendernamen des Font und nicht den Dateinamen des Font angeben. Es ist empfehlenswert, bevor man einen Font installiert sich davon zu überzeugen, daß dieser noch nicht installiert ist,

BEISPIEL

```
•FontVorh('Brush Script Mt',SpeiFont)
•Wenn([SpeiFont=0])•FontInst(c:\windows\system\Brushci.ttf)
```

SIEHE AUCH •**FontInst**(), •**FontLösch**()

Alias: (Fonction) Befehl: Funktion

Zeichnen der Kurve einer Funktion

SYNTAX •**Funktion(*Fu,VarName,Vi,Vf,Inkrement,TabNname,Stufen*)**

BESCHREIBUNG Zeichnet eine Kurve, die eine Funktion darstellt.

Fu	:	Ausdruck einer Funktion in der Form y=f(x)
VarName	:	Name der Funktionsvariablen
Vi	:	Anfangswert der Funktionsvariablen
Vf	:	Endwert der Funktionsvariablen
Inkrement	:	Inkrement (Veränderungswert) der Funktions-variablen um von ***Vi*** bis ***Vf*** zu gelangen
TabName	:	Name der Tabellenvariablen, in der die Koordinaten der Kurvenpunkte gespeichert werden
Stufen	:	Ein Wert der angibt, wieviele Punkte der Kurve ausgegeben werden sollen (jeder x-te Punkt).

ANMERKUNG Die Darstellung der Kurve, die die Funktion repräsentiert, erfolgt mit Bezugnahme auf die Achsen, die mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man als vorletzten Parameter den Namen einer Tabelle angibt, so werden die Koordinaten der Punkt der Kurve in dieser Tabelle gespeichert. Diese muß zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor der Anzahl der Punkte, die gespeichert werden sollen. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•Funktion(y=sin(Teta),Teta,0,6.3,0.1)
•TabDef(Tab1,200,2) •TabFüllen(Tab1,0)
•Funktion(y=2*x^2,x,-2,2,0.1,Tab1)
```

SIEHE AUCH •**Ableitung**(), •**Integral**(), •**FunktionPara**(), •**AbleitPara**(), •**IntegralPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (Fonctpara) Befehl: FunktionPara

Zeichnen der Kurve einer Parameterfunktion

SYNTAX •**FunktionPara(*f(t),g(t),VarName,Vi,Vf,Inkr,TabName,Stufen*)**

BESCHREIBUNG Zeichnet eine Kurve, die eine Parameterfunktion darstellt.

f(t),g(t)	:	Gleichung der Parameterfunktion
VarName	:	Name der Funktionsvariablen
Vi	:	Anfangswert der Funktionsvariablen
Vf	:	Endwert der Funktionsvariablen
Inkr	:	Inkrement (Veränderungswert) der Funktions-variablen, um von ***Vi*** bis ***Vf*** zu gelangen
TabName	:	Name der Tabellenvariablen, in der die Koordinaten der Kurvenpunkte gespeichert werden
Stufen	:	Ein Wert der angibt, wieviele Punkte der Kurve ausgegeben werden sollen

ANMERKUNG Die Darstellung der Kurve, die die Parameterfunktion repräsentiert, erfolgt mit Bezugnahme auf die Achsen, die mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man als vorletzten Parameter den Namen einer Tabelle angibt, so werden die Koordinaten der Punkt der Kurve in dieser Tabelle gespeichert. Diese muß zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor und er gibt die Anzahl der Punkte an, die gespeichert werden sollen. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•FunktionPara(x=sin(5*Teta),y=cos(3*Teta),Teta,0,6.3,0.05)
•TabDef(Tab1,200,2) •TabFüllen(Tab1,0)
•FunktionPara(x=4*(1+cos(t))*cos(t)+2,y=4*(1+cos(t))*
                              sin(t),t,0,6.3,0.1,Tab1)
```

SIEHE AUCH •**Funktion**(), •**Ableitung**(), •**Integral**(), •**AbleitPara**(), •**IntegralPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (Peint) Befehl: Füllen

Ausfüllen einer Fläche mit einer bestimmten Farbe

SYNTAX •**Füllen(*X,Y,Stil,Farbe,Rahmenfarbe*)**

BESCHREIBUNG Füllt eine Fläche ausgehend vom Punkt mit den Koordinaten **X,Y** mit der Farbe **Farbe** aus.

X,Y	:	Koordinaten des Ausgangspunktes beim Füllen
Stil	:	Füllstil (0 bis 6)
Farbe	:	Farbe
Rahmenfarbe	:	Farbe der Ränder (optional)

Stil		**Füllstil**
0	=	Füllen mit der aktuellen Füllfarbe
1	=	Füllen mit einer einheitlichen Farbe
2	=	Füllen mit einem horizontalen Raster
3	=	Füllen mit einem vertikalen Raster
4	=	Füllen mit nach links gerichteten diagonalem Raster
5	=	Füllen mit nach rechts gerichteten diagonalem Raster
6	=	Füllen mit einem karierten Raster

ANMERKUNG Dieser Befehl stellt zwei Typen von Füllen zur Verfügung:

- Füllen unter Bezugnahme auf die Farbe des Ausgangspunktes
  (In diesem Fall gibt man keine Rahmenfarbe an.)
- Füllen innerhalb einer Kontur
  (In diesem Fall gibt man eine Rahmenfarbe, die Konturfarbe, an).

BEISPIEL

```
•Füllen(320,200,1,10)
```

SIEHE AUCH •**Füllfarbe**()

Alias: (Brosse) Befehl: Füllfarbe

Definieren der Füllfarbe

SYNTAX •**Füllfarbe(*Index*)** oder •**Füllfarbe(*R,G,B*)**

BESCHREIBUNG Definiert die Füllfarbe durch die Angabe eines Index der Farbpalette oder durch die Komponenten Rot, Grün, Blau

Index : Index der Farbe in der Palette

R,G,B : Komponenten Rot, Grün, Blau der Farbe

ANMERKUNG Wenn die Farbe, die durch die Komponenten **R, G, B** bestimmt wird, in der Palette nicht existiert, dann ist die ausgewählte Farbe eine zusammengemischte Farbe.

BEISPIEL

```
•Füllfarbe(14)
•Füllfarbe(128,128,0)  ••ergibt die Farbe braun
```

SIEHE AUCH •**Pinsel**()

Alias: (PcursG) Befehl: GCursPos

Ermitteln der Position des Grafikcursors

SYNTAX •**GCursPos(*X,Y*)**

BESCHREIBUNG Ermittelt die Position des Grafikcursors und speichert sie in den Koordinaten ***X,Y.***

X,Y : Koordinaten des Grafikcursors

BEISPIEL

```
•Nadel(320,200,100,45)
•GCursPos(CurX,CurY)
•Nadel([CurX],[CurY],120,90)
```

SIEHE AUCH •**TCursPos**()

Alias: (ImgFmt) Befehl: Grafikformat

Definieren des aktuellen Grafikformates

SYNTAX •**GrafikFormat(*Fmt*)**

BESCHREIBUNG Definition des Grafikformates eines Bildes, das mit den Befehlen •**Bild**() oder •**Objekt**() geladen wird

Fmt : Grafikformat (zum Beispiel PCX)

ANMERKUNG BMP ist das Standardformat.

Folgende Formate werden unterstützt:

BMP	Windows-Bitmap (1,4,8 und 24 bit)
GIF	Compuserve Gif (4 und 8 bit)
PCX	ZSoft PCX (1,4 und 8 bit)
TGA	Targa (16,24 und 32 bit)
TIF	Tiff (1,4,8 und 24 bit)
JPG	JPEG (8 und 24 bit)

BEISPIEL

```
•GrafikFormat(PCX)
•Objekt(Foto,200,300)
```

SIEHE AUCH •**Objekt**(), •**Bild**(), •**Capt**()

Alias: (Majusc) Befehl: GrBuchst

Umwandeln in Großbuchstaben

SYNTAX

- **GrBuchst(+)** oder
- **GrBuchst(-)** oder
- **GrBuchst(*Quell,Ziel*)**

BESCHREIBUNG Dieser Befehl ermöglicht, die Anweisung zur Umwandlung der Mitteilung des Schülers in Großbuchstaben festzulegen oder eine Zeichenkette in Großbuchstaben zu konvertieren.

- ***GrBuchst(+)*** : Aktiviert die Umwandlung in Großbuchstaben (Standardeinstellung)
- ***GrBuchst(-)*** : Desaktiviert die Umwandlung
- ***GrBuchst(Quell,Ziel)*** : Konvertieren der ***Quelle*** in Großbuchstaben in ***Ziel***

BEISPIEL

```
•GrBuchst(Meldung,MeldGr)
```

SIEHE AUCH •**Sonderzeichen**(), •**KlBuchst**(), •**ZeiSetzung**()

Alias: (Hyp) Befehl: Hyp

Setzen eines Hypertext-Wortes

SYNTAX •**Hyp(*Wort,TextFarbe,HGrundFarbe,ZKette,Name,Cursor*)**

BESCHREIBUNG Setzt ein Hypertext-Wort und macht es für einen Maus-Klick aktiv

Wort	:	Hypertext-Wort
TextFarbe	:	Farbe des Hypertext-Wortes
HGrundFarbe	:	Farbe des Hintergrundes des Hypertext-Wortes
ZKette	:	Parameter, der EGO-Befehle enthalten kann
Name	:	Name, welcher der abgegrenzten Zone des Hypertext-Wortes gegeben wurde
Cursor	:	Nummer des mit dem Hypertext-Wort verbundenen Cursors

ANMERKUNG Alle Parameter außer Wort sind frei wählbar. Wenn ein Klick auf ein Hypertext-Wort erfolgt, werden 4 Systemvariable: Die Variable ***ewControle***, beinhaltet ***ewHyperControle*** , diese wiederum beinhaltet den Parameter ***Name***, wenn dieser existiert.

Die Variable ***ewHyperTexte*** ist eine Variable Liste, welche 2 Wörter enthält, deren Trennzeichen das | (vertikaler Strich) ist. Das erste Wort ist das angeklickte Hypertext-Wort, das zweite Wort ist der Parameter ***ZKette***, wenn dieser existiert. Dieser Parameter ***ZKette*** kann ein Text sein oder aus EGO-Befehlen bestehen.

Vergessen Sie nicht, vor dem Anzeigen eines Textes, der Hypertext-Worte beinhaltet, den Befehl •**HypLösch**() anzugeben.

BEISPIEL

```
•HypLösch()
Das ist ein •Hyp(Hypertext,236,255)-•Hyp(Beispiel)
•Eingabe()
Das angeklickte Wort = [chamot(ewHypertexte,'|')]
```

SIEHE AUCH •**HypLösch**(), •**HypStil**(), •**TasteInfo**(),

Alias: (SupHyp) Befehl: HypLösch

Löschen von Hypertext-Wörtern

SYNTAX ***•HypLösch()***

BESCHREIBUNG Löscht alle anwählbaren Zonen, die mit Hypertext-Wörtern verbunden sind.

ANMERKUNG Vergessen Sie nicht, diesen Befehl anzugeben, bevor Sie einen Text anzeigen, der Hypertext-Wörter enthält.

BEISPIEL

```
•HypLösch()
Das ist ein •Hyp(Hypertext,236,255)-•Hyp(Beispiel)
•Eingabe()
Das angeklickte Wort = [chamot(ewHypertexte,1,'|')]
```

SIEHE AUCH •**Hyp**(), •**HypStil**(),

Alias: (HypStil) Befehl: HypStil

Definieren des Stils für Hypertext-Worte

SYNTAX •**HypStil(*TextFarbe,HGrundFarbe,Cursor*)**

BESCHREIBUNG Definieren der Textfarbe, der Hintergrundfarbe und der Cursorart für Hypertext-Worte

TextFarbe	:	Farbe des Hypertext-Wortes
HGrundFarbe	:	Hintergrundfarbe des Hypertext-Wortes
Cursor	:	Nummer der Art des Cursors, der mit dem Hypertextwort verbunden ist

BEISPIEL

```
•HypLösch()
•HypStil(254,236,6)
Das ist ein •Hyp(Hypertext,236,255)-•Hyp(Beispiel)
•Eingabe()
Das angeklickte Wort = [chamot(ewHypertexte,1,'|')]
```

SIEHE AUCH •**Hyp**(), •**HypLösch**()

Alias: (Integrale) Befehl: Integral

Zeichnen der Kurve des Integrals einer Funktion

SYNTAX •**Integral(*Funktion,VarName,Va,Vb,Inkrement,TabName,Stufen*)**

BESCHREIBUNG Zeichnet die Kurve des Integrals einer Funktion

Funktion	:	Gleichungen der Funktion (y=f(x))
VarName	:	Name der Funktionsvariablen
Vi	:	Anfangswert der Funktionsvariablen
Vf	:	Endwert der Funktionsvariablen
Inkrement	:	Inkrement (Veränderungswert) der Variablen, um vom Anfangswert zum Endwert zu kommen
TabName	:	Name der Tabellenvariablen, in der die Koordinaten des Graphen gespeichert werden
Stufen	:	Ein Wert der angibt, wieviele Punkte der Kurve ausgegeben werden sollen (Divisor)

ANMERKUNG Die Darstellung der Kurve, die das Integral der Funktion repräsentiert, erfolgt unter Bezugnahme auf die Achsen, die zuvor mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man den Namen einer Tabelle als vorletzten Parameter angibt, so werden die Koordinaten des Graphen darin gespeichert. Die Tabelle muß aber zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor, der die Anzahl der zu speichernden Punkte bestimmt. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•Integral(y=sin(Teta),Teta,0,6.3,0.1)
•TabDef(Tab1,200,2)•TabFüllen(Tab1,0)
•Integral(y=2*x,x,-2,2,0.1,Tab1)
```

SIEHE AUCH •**Funktion**(), •**Ableitung**(), •**FunktionPara**(), ,..•**AbleitPara**(), •**IntegralPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (IntegPara) Befehl: IntegralPara

Zeichnen der Kurve des Integrals einer Parameterfunktion

SYNTAX •**IntegralPara(*f(t),g(t),VarName,Va,Ve,Inkr,TabName,Stufen*)**

BESCHREIBUNG Zeichnet die Kurve des Integrals einer Parameterfunktion

f(t),g(t)	:	Gleichungen der Parameterfunktion
VarName	:	Name der Funktionsvariablen
Va	:	Anfangswert der Funktionsvariablen
Ve	:	Endwert der Funktionsvariablen
Inkr	:	Inkrement (Veränderungswert) der Variablen, um vom Anfangswert zum Endwert zu kommen
TabName	:	Name der Tabellenvariablen, in der die Koordinaten der Kurve gespeichert werden
Stufen	:	Ein Wert der angibt, wieviele Punkte der Kurve ausgegeben werden sollen

ANMERKUNG Die Darstellung der Kurve, die das Integral der Parameterfunktion repräsentiert, erfolgt unter Bezugnahme auf die Achsen, die zuvor mit dem Befehl •**AchsenDef**() definiert wurden. Wenn man den Namen einer Tabelle als vorletzten Parameter angibt, so werden die Koordinaten des Graphen darin gespeichert. Die Tabelle muß zuvor mit dem Befehl •**TabDef**() definiert worden sein. Der letzte Parameter ist der Divisor, der die Anzahl der zu speichernden Punkte bestimmt. Wird kein Divisor angegeben, so werden alle Punkte gespeichert.

BEISPIEL

```
•AchsenDef(100,200,40,40)
•IntegralPara(x=cos((t/2)^2),y=sin((t/2)^2),t,0,12,0.05)
•TabDef(Tab1,200,2)•TabFüllen(Tab1,0)
•IntegralPara(x=cos((t/2)^2),y=sin((t/2)^2),t,0,9,0.05,tab1)
```

SIEHE AUCH •**Funktion**(), •**Ableitung**(), •**Integral**(), •**FunktionPara**(), •**AbleitPara**(), •**AchsenDef**(), •**TabDef**()

Alias: (ParPol) Befehl: IPS

Umwandeln eines in Klammern stehenden Antwortmodells in die Inverse Polnische Schreibweise

SYNTAX

- **IPS(*AMKlammer,AMPoln*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, ein Antwortmodell, welches in Klammern geschrieben ist, in die inverse polnische Schreibweise umzuwandeln, damit es auch unter •**Xfi**() verwendbar ist.

AMKlammer	:	Name der Zeichenkette, welche das Antwort-modell enthält
AMPoln	:	Name der Zeichenkette, welche das umgewandelte Antwortmodell enthält

ANMERKUNG Dieser Befehl kann auch dazu benutzt werden, um Zeichenketten die Antwortmodelle enthalten, in Antwortmodelle zu konvertieren, wie sie in den Analyseblöcken vorkommen.

BEISPIEL

```
•[Par:='POLYGON+(FIGUR*GESCHLOSSEN)']
•IPS(Klammer,Poln)
```

SIEHE AUCH

- **PalKopie**()

Alias: (Minusc) Befehl: KlBuchst

Umwandeln in Kleinbuchstaben

SYNTAX

- **KlBuchst(+)** oder
- **KlBuchst(-)** oder
- **KlBuchst(*Quelle,Ziel*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, eine Umwandlung der Mitteilung des Schülers in Kleinbuchstaben festzulegen oder eine Zeichenkette in Kleinbuchstaben zu konvertieren.

- ***KlBuchst(+)*** : Aktiviert die Umwandlung
- ***KlBuchst(-)*** : Desaktiviert die Umwandlung (Standardeinstellung)
- ***KlBuchst(Quelle,Ziel)*** : Konvertieren des Quelltextes in Kleinbuchstaben in den Zieltext

BEISPIEL

```
•KlBuchst(Nachricht,NachrKlein)
```

SIEHE AUCH •**Sonderzeichen**(), •**GrBuchst**(), •**ZeiSetzung**()

Alias: (FenCom) Befehl: KomFen

Definieren eines Kommentarfensters

SYNTAX •**KomFen(*Xh,Yh,Breite,Höhe,Textfarbe,Hgrundfarbe*)**

BESCHREIBUNG Erzeugt und öffnet ein Kommentarfenster mit dem Ursprung ***Xh,Yh***, der Breite ***Breite*** und der Höhe ***Höhe***. Der Text, der dort angezeigt werden kann, hat die Farbe ***Textfarbe***, der Hintergrund des Fensters hat die Farbe ***Hgrundfarbe***.

Xh,Yh	:	Koordinaten der oberen linken Ecke des Fensters
Breite	:	Breite des Fensters
Höhe	:	Höhe des Fensters
Textfarbe	:	Farbe des Textes im Fenster
Hgrundfarbe	:	Hintergrundfarbe des Fensters

ANMERKUNG Alle Koordinaten werden in Pixel ausgedrückt. Damit der Inhalt eines Fensters beim Öffnen nicht gelöscht wird, darf man keine Hintergrundfarbe angeben.

Um ein Kommentarfenster zu desaktivieren (Rückkehr zum Vollbildschirm ohne ihn zu löschen), ist der Befehl •**KomFen**() zu verwenden (ohne Angabe von Parametern zwischen den Klammern).

Ein Kommentarfenster kann im Textteil oder Kommentarteil jederzeit definiert werden.

Der Befehl •**Pos**(x,y) bezieht sich auf das gerade aktuelle Kommentarfenster.

BEISPIEL

```
•KomFen(100,400,200,200,5,10)
```

SIEHE AUCH •**AntwFen**(), •**Pos**()

Alias: (Precis) Befehl: Komma

Festlegen der Anzeigegenauigkeit von Zahlen

SYNTAX •**Komma(*DezimalA,DezimalB*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, die Anzeigegenauigkeit von Zahlen zu präzisieren, d.h. die Anzahl an Ziffern, die nach dem Komma angezeigt werden sollen (sofern es welche gibt), ebenso wie die Genauigkeit von Berechnungen.

DezimalA	:	Genauigkeit der Anzeige: (Zahl zwischen 0 und 15)
DezimalB	:	Genauigkeit der Berechnung: (Zahl zwischen 0 und 15)

ANMERKUNG Die Grundeinstellung für die Anzeige ist 2, für die Berechnung 15.

SIEHE AUCH •**Nullen**()

Alias: (ConvCoord) Befehl: KoordKonv

Konvertieren von relativen Koordinaten in ein Windows-Fenster

SYNTAX •**KoordKonv(*FenQuelle,xq,yq,FenZiel,VarX,VarY*)**

BESCHREIBUNG Erlaubt das Konvertieren von relativen Koordinaten von einem Fenster in ein anderes Fenster.

FenQuelle	:	Nummer des Fensters der Original-Koordinaten
xq,yq	:	Koordinaten des Originals
FenZiel	:	Nummer des Fensters, in das die Original-Koordinaten konvertiert werden
VarX,VarY	:	Name der Variablen, die die konvertierten Koordinaten übernehmen

BEISPIEL

[nFen:=ewSourisFen, MX:= ewsourisX, MY:=ewsourisY]

```
•KoordKonv([nFen],[MX],[MY],0,CX,CY)
```

SIEHE AUCH •**Fen**(), •**FenDef**(), •**FenFix**(), •**FenList**(), •**FenPos**(), •**FenLösch**() •**FenInfo**()

Alias: (Cercle) Befehl: Kreis

Zeichnen eines Kreises

SYNTAX •**Kreis(*Xc,Yc,R*)**

BESCHREIBUNG Zeichnet einen Kreis mit dem Mittelpunkt ***Xc,Yc***, und dem Radius ***R.***

Xc,Yc	:	Koordinaten des Kreismittelpunkts
R	:	Radius des Kreises

ANMERKUNG Der Kreis wird in der Farbe, dem Modus und dem Stil des aktuellen Pinsels gezeichnet.

BEISPIEL

```
•Kreis(320,200,100)
```

SIEHE AUCH •**EllipseVoll**(), •**Ellipse**(), •**Bogen**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Droite) Befehl: Linie

Zeichnen einer Linie

SYNTAX •**Linie(*Xd,Yd,Xf,Yf*)**

BESCHREIBUNG Zeichnet eine Linie, die durch die Koordinaten ***Xd,Yd,Xf,Yf*** definiert wird.

Xd,Yd,Xf,Yf : Koordinaten der Linie

ANMERKUNG Die Zeichnung erfolgt in der Farbe, dem Modus und dem Stil des Pinsels, der gerade aktuell ist.

BEISPIEL

```
•Linie(100,150,400,200)
```

SIEHE AUCH •**Nadel**(), •**Vektor**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Scur) Befehl: MausCurs

Definieren des Mauscursor-Typ

SYNTAX •**MausCurs(*Typ*)**

BESCHREIBUNG Ändert die Form des Mauscursors. Standardmäßig hat der Mauscursor die Form eines Pfeils (Typ 1). ***Typ*** kann einen Wert zwischen 0 und 11 annehmen.

Typ	Cursor
0	= Kein Cursor
1	= Pfeil
2	= Textcursor
3	= Sanduhr
4	= Kreuz
5	= vertikaler Pfeil
6	= Pfeile in vier Richtungen
7	= Rechteck
8	= Nordwest-Südost-Pfeil
9	= Nordost-Südwest-Pfeil
10	= Ost-West-Pfeil
11	= Nord-Süd-Pfeil

ANMERKUNG Die vordefinierten Cursor haben eine Nummer zwischen 0 und 11. Die persönlichen Cursor, welche mit dem Befehl •**MausCursB**() erstellt werden, werden mit den Werten zwischen 20 und 148 bezeichnet (Man kann somit bis zu 128 persönliche Cursor erstellen). Standardmäßig nimmt der Cursor den Typ 1 ein, wenn er auf eine Taste trifft. Alle nach dem Befehl •**MausCurs**(**Typ,+**) haben den Cursortyp ***Typ***. Nach dem Befehl •**MausCurs(Typ,-)** erscheinen alle existierenden Tasten mit dem Standard-Cursor. Nach dem Befehl •**MausCurs**(**Typ,*)** wird der Cursortyp ***Typ*** allen bereits existierenden und auch künftigen Tasten zugewiesen. Will man einen Cursor einer bestimmten Taste zuweisen, ist es notwendig, den Befehl •**MausCurs(Typ,VarName**) auszuführen, wobei ***VarName*** der Name der verbundenen Variablen der Taste ist.

BEISPIEL

```
•MausCurs(3)
•MausCurs(22)
•MausCurs(4,Taste1)
```

SIEHE AUCH •**MausCursB**(), •**MausPos**(), •**MausZone**()

Alias: (ScurImg) Befehl: MausCursB

Festlegen eines persönlichen Mauscursors

SYNTAX •**MausCursB(*CursNum,x,y,Breite,Höhe,Cx,Cy,IndW,IndS*)**

BESCHREIBUNG Erstellt einen Mauscursor, der aus einer Grafik besteht.

CursNum	:	Nummer des Cursors (zwischen 20 und 148)
x,y	:	Linke obere Ecke der Grafik
Breite,Höhe	:	Breite und Höhe des Cursors
Cx,Cy	:	Sensibler Punkt des Cursors. Cx und Cy geben die Verschiebung von x und y in Bezug auf die linke obere Ecke des Cursor-Rechteckes an
IndW,IndS	:	Index jener Grafikfarbe, welche man weiß und schwarz im Cursor haben möchte

ANMERKUNG Ein Mauscursor setzt sich aus zwei Teilen zusammen: Maske und Cursor. Die Maske und der Cursor müssen horizontal Seite an Seite angezeigt werden. Die Maske erlaubt die Definition der transparenten und nicht transparenten Teile des Cursors. Die Teile der Maske sind transparent, die schwarzen Teile jedoch undurchsichtig. Die weißen und schwarzen Teile des Cursors erscheinen weiß oder schwarz, wenn sie von Teilen der Maske überlagert werden, wenn nicht, erscheinen sie invers im Hintergrund. Die Breite und Höhe eines Cursors kann bis zu 32 Pixel aufweisen. Ein Mauscursor kann nur weiß oder schwarz sein, aber seine repräsentative Grafik kann mit oder ohne Farben erstellt werden, unter der Bedingung, daß der Farbindex der Farben, welche weiß oder schwarz definieren, im Befehl vorkommt.

BEISPIEL

```
•BschLösch(255)•Füllfarbe(236)•FlächeBH(0,0,32,32)
•Pinsel(255)•Pinselstil(3)
•Linie(15,0,15,11)•Punkt(15,15)•Linie(15,20,15,32)
•Linie(0,15,11,15)•Punkt(15,15)•Linie(20,15,32,15)
••MausCursB(20,0,0,32,32,15,15,255,236)
•MausCurs(20)•Pause()
•BschLösch(236)•Pinsel(255)•Pinselstil(1)
•Linie(47,0,47,11)•Punkt(47,15)•Linie(47,20,47,32)
•Linie(32,15,43,15)•Punkt(47,15)•Linie(52,15,64,15)
•MausCursB(21,0,0,32,32,15,15,255,236)
•MausCurs(21)
```

SIEHE AUCH •**MausCurs**(), •**MausPos**(), •**MausZone**()

Alias: (Sbtn) Befehl: MausKlick

Simulieren eines Mausklicks

SYNTAX

•**MausKlick(*TasteNum,Aktion*)** oder

•**MausKlick(+)** oder

•**MausKlick(-)**

BESCHREIBUNG Simuliert einen Klick, Doppelklick, anhaltenden Klick oder einen losgelassenen Klick.

•**MausKlick(+)** : Aktiviert die Maus-Aktionen (Standardeinstellung)

•***MausKlick***(-) : Desaktiviert die Maus-Aktionen

TasteNum **Nummer der aktivierten Maustaste**
- 1 = links
- 2 = rechts
- 3 = Mitte

Aktion **Art des Klicks vor der Simulation**
- 1 = Klick
- 2 = Doppelklick
- 3 = angehaltener Kick
- 0 = losgelassener Klick

BEISPIEL

```
•Extern(calc.exe,11,200,200,1)
•Pause(1)
•MausPos(440,400)•Pause(1)•MausKlick(1,1)•Pause(1) ..4
•MausPos(570,475)•Pause(1)•MausKlick(1,1)•Pause(1) ..+
•MausPos(530,360)•Pause(1)•MausKlick(1,1)•Pause(1) ..9
•MausPos(610,475)•Pause(1)•MausKlick(1,1)•Pause(1) ..=
•Pause(2)
•MausPos(230,205)•MausKlick(1,3)•MausPos(50,50)
••versetzt das Rechner-Fenster
•Pause(2)•MausPos(25,55)•MausKlick(1,2)
••Mausklick wird beendet
•Pause(2)•MausPos(25,55)•MausKlick(1,2)
••Doppelklick auf Menü-System
•Pause(2)
•Ende()
```

SIEHE AUCH •**MausPos**()

Alias: (Spos) Befehl: MausPos

Positionieren des Mauscursors

SYNTAX •**MausPos(*x,y*)**

BESCHREIBUNG Positioniert den Mauscursor am Punkt mit den Koordinaten ***x*** und ***y***;

x,y : Koordinaten der Mausposition

x und ***y*** werden in Pixel ausgedrückt.

BEISPIEL

•MausPos(320,200)

SIEHE AUCH •**MausCurs**()

Alias: (SLimite) Befehl: MausZone

Definieren der Bildschirmzone, innerhalb der die Maus bewegt werden kann

SYNTAX •**MausZone**(***Xhg,Yhg,Xbd,Ybd***)

BESCHREIBUNG Definition der Zone, innerhalb der der Mauscursor bewegt werden kann.

Xhg,Yhg : linke obere Ecke der Mauszone

Xbd,Ybd : rechte untere Ecke der Mauszone

ANMERKUNG Standardmäßig kann die Maus am gesamten Bildschirm bewegt werden.

Um wieder den gesamten Bildschirm zum Mausbereich zu machen, muß lediglich der Befehl •**MausZone**() ohne Parameter angegeben werden.

BEISPIEL

```
•MausZone(320,200,500,300)
```

SIEHE AUCH •**MausCurs**(), •**MausPos**()

Alias: (MCI) Befehl: MCI

Setzen von Befehlen zur MULTIMEDIA-Steuerung

SYNTAX •**MCI(*Befehl,Var*)**

BESCHREIBUNG Überträgt einen Steuerbefehl an eine periphere Multimedia Komponente z.B. an eine Sound- oder Videokarte.

Befehl : Kette von Befehlen

Var : Name der Variablen, die die Parameter der Rückmeldung enthält, wenn der MCI Befehl eine Rückmeldung provoziert.

ANMERKUNG Bezugnahme auf die Liste der MCI Befehle von Windows, unter Einbindung der Hardware (Audio-Karte, Digitalisierungskarte, Video für Windows ...). Für die WAV und MID Sequenzen bietet EGO einen übergeordneten Befehl LAUTSTÄRKE an, welcher es erlaubt, die Lautstärke auf dem rechten und linken Kanal zwischen den Werten 0 und 54000 zu regeln. Im 256 Farbmodus wird die Anpassung an die laufende Farbpalette einer Videosequenz erhalten, indem der folgende MCI-Befehl ausgeführt wird:

•**MCI(*setvideo alias Palette handle to [sys(6)]*)**

BEISPIEL

```
•MCI(open d:\son\Applaus.wav alias bravo)
•MCI(capability device type,TypePeriph)
•MCI(volume bravo 5000)
•MCI(play bravo)
```

Alias: (ModListe) Befehl: ModListe

Gibt die Liste der Module eines EGO Prozesses wieder

SYNTAX •**ModListe(*Prozess,VarName,Trennz*)**

BESCHREIBUNG Gibt die Liste der Module im Speicher eines EGO Prozesses wieder

Prozess	:	Nummer des EGO Prozesses (zwischen 0 und 16)
VarName	:	Name einer Variablenkette, die die Namen der Module aus dem Speicher übernimmt
Trennz	:	Trennzeichen zur Verwendung für die Modulliste. (Zeichen I standardmäßig)

ANMERKUNG Der Prozeß 0 ist der Grundprozeß, die Prozesse von 1 bis 16 sind die Aufgaben aus dem Lauf der Ausführung. Die Namen der Module aus dem Speicher eines EGO Prozesses werden wiedergegeben in der Zeichenkette ***VarName*** mit dem Zeichen ***Trennz*** als Trennzeichen.

BEISPIEL

```
•ModListe(0,PName0) ••Liste der Co-Module im Speicher
•ModListe(3,PName3) ••Liste der Co-Aufgaben des Prozesses 3
```

SIEHE AUCH •**ModStat**(), •**CoMod** (), •**CoAufgabe**

Alias: (NomMod) Befehl: ModName

Übernimmt den Namen des gerade in Ausführung befindlichen Moduls

SYNTAX •**ModName (*Name*)**

BESCHREIBUNG Dieser Befehl übernimmt den Namen des gerade in Ausführung befindlichen Moduls.

Name : Name der Variablen, welche den Namen des Moduls enthält

BEISPIEL

```
••Speichern des Namens des aktiven Moduls
•ModName(ModName)
```

SIEHE AUCH •**VarName**()

Alias: (ModParam) Befehl: ModParam

Ermitteln der Parameter des vorhergehenden Moduls

SYNTAX •**ModParam(*Name,Var1,Var2,...,VarN*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln der übernommenen Parameter, im Zusammenhang mit einem der folgenden Befehle:

•**Weiter**(), •**CoMod**(), •**ModVerb**() oder •**Aufgabe**()

Name	:	Name der numerischen Variablen, welche die Anzahl der Parameter enthält, die vom Modul übernommen werden
Var1...VarN	:	Name der Zeichenketten-Variablen, welche die Parameter enthalten, die vom Modul übernommen werden

ANMERKUNG Alle erhaltenen Parameter werden ausschließlich in Zeichen-kettenvariablen gespeichert. Es können alle Arten von Parametern übernommen werden: Texte, Zahlen, EGO-Befehle.

Die maximale Anzahl der erhaltenen Parameter ist 32.

BEISPIEL

```
•ModParam(ParamNum,_P1,_P2,_P3)
```

SIEHE AUCH •**Weiter**(), •**CoMod**(), •**ModVerb**(), •**Aufgabe**()

Alias: (ModStat) Befehl: ModStat

Gibt den Zustand der Modules von einem EGO-Prozess wieder

SYNTAX •**ModStat(*Prozess,ModName,VarName*)**

BESCHREIBUNG Gibt den Zustand der Modules von einem EGO-Prozess wieder

Prozess	:	Name des EGO Prozesses (zwischen 0 und 16)
ModName	:	Modulname
VarName	:	Name einer numerischen Variablen, die den Zustand eines Modules übernimmt.

ANMERKUNG Der Prozeß 0 ist der Grundprozess, die Prozesse von 1 bis 16 sind die Aufgaben aus dem Lauf der Ausführung. Der Zustand der Module wird wiedergegeben in der Variable ***VarName***.

Dieser Befehl kann Werte zwischen 0 und inclusive 7 übernehmen:
0 = Name des invaliden Modules
1 = Aktiv laufende Modul (jenes, das •**ModStat()** aufgerufen hat)
2 = Aktives Modul in der Liste der aktiven Prozesse
3 = Inaktives Modul, unter dem es Co-Module oder Co-Aufgaben gibt
4 = Blockertes Modul durch •**AufgPause**()
5 = Beendetes Modul im Zustand der Löschung aus dem Speicher
6 = Unterbrochenes Modul durch •**Pause**()
7 = Modul an einem Interventionspunkt •**Eingabe**()

BEISPIEL

```
•ModStat(0,c:\beispiele\CoMod3,Zustand)
```

SIEHE AUCH •**ModListe**(), •**AufgPause**(), •**Pause**() •**Eingabe**()

Alias: (ExecMod) Befehl: ModVerb

Erzwungene Verbindung mit einem Modul

SYNTAX •**ModVerb**(***Modulname,+/-,p1,...,pN***)

BESCHREIBUNG Lädt und führt ein Modul aus, ausgehend von einer Aufgabe oder einem Co-Modul.

Modulname	:	Name des Moduls
+/-	:	Option zur Speicherung der Anhäufung der Übersetzungen
p1,...pN	:	Parameter beim Übergang zum Modul

ANMERKUNG Dieser Befehl, der ausgehend von einer Aufgabe oder einem Co-Modul ausgeführt wird, beendet alle lokalen Aufgaben und laufenden Co-Module, lädt und führt das Modul Modulname aus. Erfolgt keine Angabe, so werden die angereicherten Übersetzungen gespeichert.

Um die Anreicherung zu löschen, muß als zweiter Parameter das Zeichen ´-´angegeben werden. Dieser Befehl ist notwendig, um Module ausgehend von Aufgaben oder Co-Modulen zu verbinden.

Das aufgerufene Modul kann die Parameter wiedererlangen, die vom vorhergehenden Modul übernommen wurden. Dies geschieht mit dem Befehl •**ModParam**(). Es können bis zu 32 Parameter übernommen werden.

BEISPIEL

```
••Verkettung mit dem Modul 2 und
••Löschen der Anhäufung der Übersetzungen
•ModVerb(Modul2,-)
```

SIEHE AUCH •**Weiter**(), •**CoMod**(), •**Ende**(), •**Aufgabe**(), •**AufgPause**(), •**AufgFort**()

Alias: (Aig) Befehl: Nadel

Zeichnen einer Nadel

SYNTAX •**Nadel(*Xo,Yo,Länge,Winkel*)**

BESCHREIBUNG Zeichnet eine Nadel mit dem Ursprung Xo,Yo und der Länge Länge angelegt im Winkel Winkel. Der Winkel wird in Grad in trigonometrischer Richtung, ausgehend von der Horizontalen, ausgedrückt.

Xo,Yo	:	Koordinaten des Ursprungs
Länge	:	Länge der Nadel
Winkel	:	Ausrichtung der Nadel

ANMERKUNG Das Zeichnen erfolgt in der Farbe, im Stil und im Modus des aktuellen Pinsels.

BEISPIEL

```
•Nadel(320,200,150,45)
```

SIEHE AUCH •**Vektor**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (FIncomp) Befehl: NichtVerst

Setzen des Pfades der Datei mit den nicht verstandenen Mitteilungen

SYNTAX •**NichtVerst(*Pfad*)** oder

•**NichtVerst(-)** oder

•**NichtVerst()**

BESCHREIBUNG Gibt den Pfad an, wo sich die Datei mit der Zeichenkette der Nichtverstandenen Antworten befindet. Diese Zeichenkette, die sich ***ewFinc.cha*** nennt, beinhaltet alle Mitteilungen des Schülers, die von keinem Antwortmodell erkannt wurden.

ANMERKUNG Erfolgt keine spezielle Angabe, so entspricht der Pfad der Datei mit den nicht verstandenen Mitteilungen dem Pfad des aktuellen Moduls.

•**NichtVerst**(-) untersagt das Speichern in der Datei der nicht verstandenen Mitteilungen.

Wenn man den Befehl •**NichtVerst**() ohne Pfad angibt, so stimmt der Pfad der Datei wieder mit dem des aktuellen Moduls überein.

BEISPIEL

```
•NichtVerst(d:\Mappe\)
```

Alias: (Zer) Befehl: Nullen

Anzeigen von Nullen am Ende von Zahlen

SYNTAX •**Nullen(+)** oder •**Nullen(-)**

BESCHREIBUNG Nach der Ausführung dieses Befehls wird die Darstellung von Dezimalzahlen durch Nullen am Ende vervollständigt. Die geforderte Präzisierung erfolgt durch den Befehl •**Komma**().

•***Nullen***(+) : aktiviert die Option

•***Nullen***(-) : desaktiviert die Option

ANMERKUNG Nach der Anzeige ist es ratsam, den Befehl wieder zu desaktivieren, da jeder Art von Parametrisierung Nullen hinzugefügt werden.

Erfolgt zum Beispiel durch •**Weiter**(**Mod[x])** eine Verbindung mit einem Modul, dessen Name ein Parameter ist, wobei •Nullen(+) noch immer aktiv ist, so versucht EGO bei einem Wert x=3 und einer Präzisierung von 2 eine Verkettung mit dem Modul 3.00, was nicht der erwartete Effekt ist.

BEISPIEL

```
•Komma(2)
[USt:=18.6]
•Nullen(+)
[USt]
•Nullen(-)
```

SIEHE AUCH •**Komma**()

Alias: (ResListeCanaux) Befehl: NwKanalAktiv

Erstellen der Liste der aktiven Kanäle

SYNTAX •**NwKanalAktiv(*TabName,VarName*)**

BESCHREIBUNG Dieser Befehl erlaubt es, in einer Tabelle ***TabName*** die Nummern der aktiven Kanäle und in ***VarName*** die Anzahl der aktiven Kanäle zu erfassen.

TabName	:	Name der Tabellen-Variablen, die die Nummern der aktiven Kanäle enthält
VarName	:	Name der numerischen Variablen, die die Anzahl der aktiven Kanäle enthält

ANMERKUNG Es ist nicht notwendig, die Tabelle TabName vor der Ausführung des Befehls •**NwKanalAktiv**() zu dimensionieren, weil diese in Funktion der Anzahl der aktiven Kanäle erfolgt..

BEISPIEL

```
•NwKanalAktiv(KanalListe,KanalAnzahl)
```

SIEHE AUCH •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(),•**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResListeConnex) Befehl: NwKanalListe

Erstellen der Liste der Verbindungen auf einem Kanal

SYNTAX •**NwKanalListe(*Kanal,Liste,Trennz,VarName*)**

BESCHREIBUNG Dieser Befehl erlaubt es, in der Zeichenkette Liste die Namen der existierenden Verbindungen auf einem Kanal zu erfassen

Kanal	:	Nummer des Verbindungskanals
Liste	:	Name der Zeichenkette, die die Liste der Namen der Verbindungen enthält
Trennz	:	Trennzeichen der Worte in Liste
VarName	:	Optional. Name der numerischen Variablen, die die Anzahl der in Liste gespeicherten Namen enthält

ANMERKUNG Es ist zwingend, ein Trennzeichen anzugeben. Wenn die Variable ***VarName*** einen negativen Wert hat, dann bedeutet dies, daß es mehr Namen gibt, als die Zeichenkette aufnehmen kann (max. 1024 Zeichen). Der absolute Wert von ***VarName*** gibt die Anzahl aller Verbindungen an und die Namen können einer nach dem anderen mit Hilfe des Befehls •**NwVerbName**() ermittelt werden.

BEISPIEL

```
•NwKanalListe(33,Liste,/)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResQuiSuisJe) Befehl: NwKarteNr

Ermitteln der Seriennummer der Netzwerkkarte

SYNTAX •**NwKarteNr(*Name*)**

BESCHREIBUNG Dieser Befehl erlaubt das Ermitteln der Seriennummer der Netzwerkkarte.

Name : Name der Zeichenketten-Variablen, die die Seriennummer der Netzwerkkarte enthält

ANMERKUNG Die Seriennummer der Netzwerkkarte ist einzigartig, sie kann den Namen oder einen Teilnamen der Verbindung enthalten

BEISPIEL

```
•NwKarteNr(Name)
•NwVerb(33,[Name])
•NwVerb(33,Spieler[Name])
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResInit) Befehl: NwInit

Initialisieren der Netzwerk-Parameter

SYNTAX •**NwInit(*Socket,Nachricht,Frist,VarName*)**

BESCHREIBUNG Dieser Befehl erlaubt das Verändern von Initialisations-Parametern des Netzwerkes.

Socket	:	Nummer des Anschlusses IPX (13107 ohne Angabe)
Nachricht	:	Nummer der an EGO übergebenen Windows-Nachricht, die eine EGO-Zeichenkette über-nommen hat (3000 ohne Angabe)
Frist	:	Frist zur Überprüfung der Ankunft einer EGO Zeichenkette
VarName	:	Name der Variablen, welche das Ergebnis der Überprüfung des Vorhandenseins eines Novell-Netzwerkes enthält (0 = Kein Netzwerk, 1 = Netzwerk ist vorhanden)

ANMERKUNG Es ist nicht notwendig, eine Netzwerkverbindung mit diesem Befehl zu beginnen. Im Normalfall sind die Werte der Grundeinstellung in der Mehrzahl der Fälle gültig. Immer, wenn eine Anwendung eine Anschlußnummer oder Nummer der Nachricht durch Zufall verwendet, ist es sehr einfach eine nicht benutzte auszuwählen.

Darüber hinaus kann dieser Befehl das Vorhandensein eines Novell-Netzwerkes überprüfen.

BEISPIEL

```
•NwInit(,,,vorhanden)
•Wenn([vorhanden=0])Kein Netzwerk vorhanden!•Ende()
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResLitMess) Befehl: NwNachrEmpf

Empfangen einer EGO-Zeichenkette

SYNTAX •**NwNachrEmpf(*Kanal,Quell,Ziel,ZKette*)**

BESCHREIBUNG Dieser Befehl erlaubt das Empfangen des Inhalts einer EGO-Zeichenkette, welche durch Quell an Ziel auf dem Kanal übertragen wurde.

Kanal	:	Nummer des Kanals, auf dem die Verbindung zustandekommt
Quell	:	Name des Gerätes, welches die EGO-Zeichen-kette übersendet
Ziel	:	Name des Gerätes, an das die EGO-Zeichen-kette übersandt wird
ZKette	:	Name der zu empfangenden Zeichenkette

ANMERKUNG Die Geräte ***Quell*** und ***Ziel*** müssen mit dem Befehl •**NwVerb**() verbunden worden sein. Um den Inhalt der Zeichenkette lesen zu können, ist es notwendig, die Nummer des Übertragungskanals der Zeichenkette zu kennen, ebenso den Namen des Senders und des Empfängers. Die Kanal-Nummer läßt sich mit dem Befehl •**NwNachrWart**() ermitteln, die Namen des Senders und des Empfängers mit dem Befehl •**NwNachrQuell**().

BEISPIEL

```
•NwNachrWart(Kanal)
•Wenn([Kanal=0])•Sprung(=-1)
•NwNachrQuell([Kanal],Quell,Ziel)
•Wenn([Quell<>'',=-3])
•NwNachrEmpf([Kanal],[Quell],[Ziel],Nachricht)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResMessOK) Befehl: NwNachrOK

Überprüfen, ob die abgesandte EGO-Zeichenkette erhalten wurde

SYNTAX •**NwNachrOK(*NumSend,VarName*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Überprüfung, ob eine abgesendete Zeichenkette am Zielort angekommen ist.

NumSend	:	Nummer der Nachricht, welche durch •**NwNachrSend**() retourniert wurde
VarName	:	Nummer der numerischen Variablen, die das Resultat der Überprüfung enthält

ANMERKUNG Wenn die Variable ***VarName*** nicht Null ist, dann wurde die Nachricht erhalten.

BEISPIEL

```
[Nachr:='Guten Tag Gerät 2']
•NwNachrSend(33,Gerät1,Gerät2,Nachricht,Zurück)
•Wenn([Zurück=0])•Sprung(=-1)
•NwNachrOK([zurück],OK)
•Wenn([OK=0])•Sprung(=-1)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResQuelMess) Befehl: NwNachrQuell

Ermitteln der Namen des Senders und Empfängers der EGO-Zeichenkette

SYNTAX •**NwNachrQuell(*Kanal,Quell,Ziel*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln des Namens des Senders und des Empfängers einer EGO-Zeichenkette, die in der Warteschlange des Kanals Kanal erhalten wird.

Kanal	:	Nummer des Verbindungskanals
Quell	:	Name der Variablen, die den Namen des Senders enthält
Ziel	:	Name der Variablen, die den Namen des Empfängers enthält

ANMERKUNG Wenn die Variable ***Quell*** leer ist, dann ist keine EGO-Zeichenkette in der Warteschlange des Kanals. Dieser Befehl wird im Wechsel mit dem Befehl •**NwNachrWart**() verwendet, der überprüft, ob eine Zeichenkette erhalten wurde und auf welchem Kanal.

BEISPIEL

```
•NwNachrWart(Kanal)
•Wenn([Kanal=0])•Sprung(=-1)
•NwNachrQuell([Kanal],Quell,Ziel)
•Wenn([Quell<>''],=-3)
•NwNachrEmpf([Kanal],[Quell],[Ziel],Nachricht)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResEnvoieMess) Befehl: NwNachrSend

Senden einer EGO-Zeichenkette

SYNTAX •**NwNachrSend(*Kanal,Quell,Ziel,ZKette,NumSend,Send*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Senden des Inhalts einer EGO-Zeichenkette an ein Gerät mit dem Namen Ziel, verbunden auf dem Kanal.

Kanal	:	Nummer des Kanals auf dem die Übertragung stattfindet
Quell	:	Name des Gerätes, das die Zeichenkette enthält
Ziel	:	Name des Gerätes, das die Zeichenkette erhält
ZKette	:	Name der zu übertragenden ZKetten-Variablen
NumSend	:	Optional. Variablenname, der die Identifikationsnummer der Übertragung enthält
Send	:	Optional. Maximale Anzahl der Versuche der Übertragung

ANMERKUNG Die Geräte ***Quell*** und ***Ziel*** müssen mit dem Befehl •**NwVerb**() verbunden worden sein. ***NumSend*** enthält einen Wert ungleich Null, wenn die Nachricht erhalten werden kann. Wenn der Wert Null ist, dann ist die Warteschlange voll. Man muß dann warten und die Übertragung wiederholen. Wenn der Wert größer Null ist, dann ist es mit Hilfe des Befehls •**NwNachrOK**() möglich, den Empfang der Zeichenkette zu überprüfen. Wenn der Parameter ***Send*** verwendet wird, führt EGO automatisch die Anzahl der Versuche durch. Wenn der Wert von ***NumSend*** kleiner 0 ist, dann hat die Übertragung nicht funktioniert.

BEISPIEL

```
[Nachr:='Gutren Tag Gerät 2]
•NwNachrSend(33,Gerät1,Gerät2,Nachr,Zurück,100)
•Wenn([Zurück<0])Die Übertragung ist gescheitert!
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResYatilMess) Befehl: NwNachrWart

Überprüft ob eine EGO-Zeichenkette in der Warteschlange ist

SYNTAX •**NwNachrWart(*Kanal*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, das Vorhandensein einer EGO-Zeichenkette in der Warteschlange zu überprüfen ebenso wie auch die Nummer des Verbindungskanals.

Kanal : Name der Variablen, die die Nummer des Verbindungskanals enthält

ANMERKUNG Wenn die Variable ***Kanal*** Null ist, dann ist keine EGO-Zeichenkette in der Warteschlange. Wenn die Variable ***Kanal*** nicht Null ist, dann beinhaltet sie die Nummer des Verbindungskanals, ein notwendiger Parameter für den Befehl •**NwNachrQuell**()

BEISPIEL

```
•NwNachrWart(Kanal)
•Wenn([Kanal=0])•Sprung(=-1)
•NwNachrQuell([Kanal],Quell,Ziel)
•Wenn([Quell<>''],=-3)
•NwNachrEmpf([Kanal],[Quell],[Ziel],Nachricht)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResConnecte) Befehl: NwVerb

Verbinden eines Gerätes auf einem Kanal

SYNTAX •**NwVerb**(***Kanal,Name***)

BESCHREIBUNG Dieser Befehl erlaubt das Verbinden eines Gerätes auf einem Netzwerkkanal

Kanal	:	Nummer des Kanals (zwischen 1 und 65535)
Name	:	Name, das dem verbundenen Gerät gegeben wurde (max. 30 Zeichen)

ANMERKUNG Wenn ein Gerät sich verbindet, dann wird eine Adresse im Netzwerk erstellt. Diese Adresse besteht aus der Kanalnummer und dem Namen, der mit Hilfe des Befehls •***NwVerb***() gegeben wurde. Der Name kann eine Folge von alphanumerischen Zeichen sein.

BEISPIEL

```
•NwInit(,,,vorhanden)
•Wenn([vorhanden=0])Es gibt kein Netzwerk!•Ende()
•NwVerb(33,Gerät1)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResNomConnect) Befehl: NwVerbName

Ermitteln des Namens einer Verbindung

SYNTAX •**NwVerbName(*Kanal,Num,Name,VarName*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln des Namens einer Verbindung.

Kanal	:	Nummer des Verbindungskanals
Num	:	Nummer des Kanals, dessen Namen man ermitteln will
Name	:	Name der Variablen, die den Namen der Verbindung enthält
VarName	:	Name der numerischen Variablen, die das Resultat enthält

ANMERKUNG Wenn die Variable ***VarName*** Null ist, dann gibt es keine Verbindung ***Num***

BEISPIEL

```
•NwKanalListe((33,Liste,/,Zurück)
•Wenn([Zurück<0],Weiter)
[i:=1,Zurück:=abs(zurück)]
•NwVerbName(33,[i],Name[i])[i:=i+1]
•Schleife([Zurück-1],=-1)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbEnde**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResFerme) Befehl: NwVerbEnde

Schließen einer lokalen Verbindung

SYNTAX •**NwVerbEnde(*Kanal,Name)***

BESCHREIBUNG Dieser Befehl ermöglicht das Schließen einer lokalen Verbindung.

Kanal : Nummer des Verbindungskanals

Name : Name des abzuschließenden Gerätes

ANMERKUNG Ein Gerät kann mehrere Verbindungen haben. Dieser Befehl ermöglicht ausschließlich das Schließen einer lokalen Verbindung

BEISPIEL

```
•NwVerbEnde(33,Gerät1)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEndeA**(), •**NwVerbEndeD**()

Alias: (ResFermeTout) Befehl: NwVerbEndeA

Schließen aller lokalen Verbindungen

SYNTAX •**NwVerbEndeA()**

BESCHREIBUNG Dieser Befehl ermöglicht das Schließen aller lokalen Verbindungen.

ANMERKUNG Ein Gerät kann mehrere Verbindungen haben. Dieser Befehl ermöglicht das Schließen aller offenen lokalen Verbindungen.

BEISPIEL

```
•NwVerbEndeA()
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeD**()

Alias: (ResFermeDist) Befehl: NwVerbEndeD

Schließen einer entfernten Verbindung

SYNTAX •**NwVerbEndeD(*Kanal,Name*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Schließen einer entfernten Verbindung.

Kanal	:	Nummer des Verbindungskanals
Name	:	Name des zu abzutrennenden Gerätes

ANMERKUNG Ein Gerät kann mehrere Verbindungen haben. Dieser Befehl ermöglicht das Schließen einer entfernten Verbindung, d.h. das abtrennen eines entfernten Gerätes.

BEISPIEL

```
•NwVerbEndeD(33,Gerät1)
```

SIEHE AUCH •**NwKanalAktiv**(), •**NwKanalListe**(), •**NwKarteNr**(), •**NwInit**(), •**NwNachrEmpf**(), •**NwNachrOK**(), •**NwNachrQuell**(), •**NwNachrSend**(), •**NwNachrWart**(), •**NwVerb**(), •**NwVerbName**(), •**NwVerbEnde**(), •**NwVerbEndeA**()

Alias: (Objet) Befehl: Objekt

Anzeigen eines Bildes

SYNTAX •**Objekt(*Name,X,Y,Modus,Operation,Alias*)**

BESCHREIBUNG Ermöglicht das Anzeigen eines Bildes.

Name	:	Name des anzuzeigenden Bildes
X,Y	:	Anzeigekoordinaten der oberen linken Ecke
Modus	:	Anzeigemodus des Bildes (siehe •**Bild**(), Standardeinstellung = 1)
Operation	:	Operation, die mit der Grafik ausgeführt werden soll (siehe •**Bild**(), Standardeinstellung = 3)
Alias	:	Name des gespeicherten Hintergrundes

BEISPIEL

```
••Anzeige eines Bildes bei Koordinaten 0,0
•Objekt(Grafik.PCX)
••Anzeige eines Bildes an den Koordinaten 100,200
•Objekt(Bild.PCX,100,200)
••Anzeige eines Bildes mit Speicherung des Hintergrundes
•Objekt(Bild.PCX,100,200,10,Hintergrund)
```

SIEHE AUCH •**Bild**(), •**GrafikFormat**(), •**BHDiv**(), •**BildDim**(), •**BildEntf**(), •**BildMax**()

Alias: (PalLum) Befehl: PalFade

Kopieren einer virtuellen Farbpalette in die aktuelle Farbpalette

SYNTAX •**PalFade(*VPalNum,Etappen*)**

BESCHREIBUNG Dieser Befehl ermöglicht es, eine virtuelle Palette in einem oder in Etappen in die aktuelle Palette zu kopieren.

VPalNum	:	Nummer der virtuellen Palette (1 bis 16)
Etappen	:	Anzahl an Etappen, in denen die virtuelle Palette in die Aktuelle kopiert wird

ANMERKUNG Der Parameter ***Etappen*** ermöglicht es, die Geschwindigkeit des Kopierens zu variieren. Je größer die Etappen sind, desto langsamer erfolgt die Kopie.

Der Parameter ***Etappen*** kann Werte zwischen 1 und 255 annehmen. Die virtuelle Palette muß mit dem Befehl •**PalKopie**() erzeugt worden sein.

BEISPIEL

```
•Objekt(Foto,,,,0)     ••laden (aber noch nicht anzeigen)
•PalKopie(1,50,150,0,Foto)
•PalFade(1,100)
```

SIEHE AUCH •**PalKopie**(), •**PalOp**()

Alias: (CopPal) Befehl: PalKopie

Kopieren der aktuellen Farbpalette in eine virtuelle Farbpalette

SYNTAX •**PalKopie(*VPalNum,IndexB,IndexE,Lum,Bildname*)**

BESCHREIBUNG Ermöglicht das Kopieren der Gesamtheit oder eines Teiles der aktuellen Farbpalette oder der Farbpalette eines geladenen Bildes in eine virtuelle Farbpalette, wobei die Helligkeit verstärkt oder vermindert werden kann.

VPalNum	:	Nummer der virtuellen Palette (1 bis 16)
IndexB	:	Anfangsindex des Palettensegmentes, das kopiert werden soll
IndexE	:	Endindex des Teils der aktuellen Palette, der kopiert werden soll
Lum	:	Prozentsatz der Helligkeit des zu kopierenden Segmentes
Bildname	:	Name des Bildes, dessen Palette man kopieren will

ANMERKUNG Das Bild muß geladen sein, bevor man die Palette kopieren kann. Der Parameter ***Lum*** gibt den Prozentsatz an, um den die Helligkeit des zu kopierenden Palettensegmentes erhöht oder vermindert werden soll. Wenn ***Lum*** negativ ist, wird die Helligkeit vermindert, ist Lum positiv, wird sie erhöht. Wenn ***Lum*** gleich 0 ist, so erfolgt keine Veränderung der Helligkeit.

Dieser Befehl ist nur bei Grafikmodi anwendbar, die 256 Farben zur Verfügung stellen.

Der Befehl beeinflußt die 20 Windows-Systemfarben nicht, die die Indizes 236 bis 255 haben.

BEISPIEL

```
•PalKopie(1,10,50,-80)
•PalKopie(2,0,235,0)
••In den Speicher geladen (nicht dargestellt)
•Objekt(Foto,,,,0)
•PalKopie(3,50,150,0,Foto)
```

SIEHE AUCH •**PalFade**(), •**Bild**(), •**Objekt**(), •**GrafikFormat**()

Alias: (OpPal) Befehl: PalOp

Operationen bei virtuellen und physischen Paletten

SYNTAX •**PalOp(*PQuelle,PZiel,IQuelleA,IQuelleE,IZiel,Op,AktionRGB*)**

BESCHREIBUNG Dieser Befehl ermöglicht:

das Kopieren eines Segmentes einer Palette in eine andere (***Op*** = 0)

das Addieren eines Segmentes einer Palette in eine andere (***Op*** = 1)

das Subtrahieren eines Segmentes einer Pal. in eine andere (***Op*** = 2)

ein fortlaufender Austausch in einem Segment einer Palette (***Op*** = 3)

PQuelle	:	Nummer der Quell-Palette (0 bis 16)
PZiel	:	Nummer der Ziel-Palette (0 bis 16)
IQuelleA	:	Index des Anfangssegmentes der Quell-Palette
IQuelleE	:	Index des Endsegmentes der Quell-Palette
IZiel	:	Ziel-Index der Zielpalette des Quellsegmentes
Op	:	Typ der Durchführungsoperation im Paletten-Segment (0 bis 3)
AktionRGB	:	Definition der RGB-Komponenten, auf die sich diese Operation bezieht 1 = Rot, 2 = Grün, 4 = Blau oder einer additiven Kombination dieser Farben: 3 wählt die 2 Komponenten Rot und Grün 5 wählt die 2 Komponenten Rot und Blau 6 wählt die 2 Komponenten Grün und Blau 7 wählt die 3 Komponenten Rot, Grün und Blau

ANMERKUNG Die Palette 0 ist die physische Palette. Im Fall einer Rotation eines Segmentes (Op = 3) ***IZiel*** ist nicht ein Index des Ziels, aber ein Index der Geschwindigkeit und der Richtung: Negativ = von Rechts nach Links, Positiv = von Links nach Rechts, darüber hinaus ist der Parameter ***AktionRGB*** ohne Bedeutung. Dieser Befehl ist nur unter dem Graphik-Modus mit 256 Farben verwendbar.

BEISPIEL

```
[c:=0,x:=1]
•Füllfarbe([c]),Fläche([x],100,[x+3],200)[x:=x+3,c:=c+1]
•Schleife(255,=-1)
•PalOp(0,0,20,85,1,3)•PalOp(0,0,100,120,-2,3)•Sprung(=)
```

SIEHE AUCH •**PalKopie**(), •**PalFade**(), •**RGBWechsel**(), •**RGBIndex**(), •**Farbskala**()

Alias: (Pause) Befehl: Pause

Pause in der Ausführung eines Moduls

SYNTAX •***Pause(Dauer)***

BESCHREIBUNG Dieser Befehl bewirkt eine Pause in der Ausführung des Moduls.

Dauer : Dauer der Pause in Sekunden

ANMERKUNG Wenn ***Dauer*** gleich 0 ist, so ist die Pause unbegrenzt und wird nur durch einen Tastendruck oder Mausklick beendet.

Wenn ***Dauer*** ein negativer Wert ist, so wird die Pause nach Ablauf der ***Dauer*** (als absoluter Wert) oder durch einen Tasten- oder Mausklick beendet.

Dauer kann eine Dezimalzahl sein.

BEISPIEL

```
•Pause(0.5)      ••eine halbe Sekunde
```

Alias: (Pinceau) Befehl: Pinsel

Definieren der Pinselfarbe

SYNTAX

- **Pinsel(*Index*)** oder
- **Pinsel(*R,G,B*)**

BESCHREIBUNG Definition der Farbe des Pinsels durch Angabe eines ***Index*** der Palette oder der Komponenten Rot, Grün, Blau

Index	:	Index der Farbe in der Palette
R,G,B	:	Rot, Grün, Blau Komponenten der Farbe

ANMERKUNG Wenn die Farbe, die durch die Komponenten ***R,G,B*** angegeben wurde, in der Palette nicht existiert, wird die der gewünschten Farbe ähnlichste Farbe der Palette ausgewählt.

BEISPIEL

```
•Pinsel(4)
•Pinsel(128,128,0)
```

SIEHE AUCH

- **Füllfarbe**()

Alias: (Pstyle) Befehl: PinselStil

Definieren des Pinselstils

SYNTAX •**PinselStil(*Stärke,Punktierung*)**

BESCHREIBUNG Definiert die Stärke und Punktierung des Pinsels.

Stärke	:	Stärke des Pinsels in Pixel
Punktierung	:	Typ der Punktierung (1 bis 6) Standardwert = 1 (durchgehend)

ANMERKUNG Die Punktierung funktioniert nur bei einer Pinselstärke von 1.

BEISPIEL

```
•PinselStil(20)
•PinselStil(1,4)
```

SIEHE AUCH •**ZeiMod**()

Alias: (Poly) Befehl: Polygon

Zeichnen eines Polygons

SYNTAX •**Polygon(*Tabellenname,+/-*)**

BESCHREIBUNG Zeichnet ein Polygon, dessen Eckkoordinaten in der Tabelle ***Tabellenname*** gespeichert werden.

Tabellenname : Name der Variablen Tabelle, die die Eckkoordinaten des Polygons enthalten

ANMERKUNG Die Tabelle ist eindimensional und enthält eine Folge von Koordinatenpaaren ***X,Y,*** mit denen die Ecken des Polygons definiert werden.

Wenn der Parameter **+** angegeben wird, so wird das Polygon mit der Farbe der aktuellen Füllfarbe ausgemalt.

Wenn der Parameter **-** gesetzt ist, dann ist das Polygon nicht geschlossen.

Die Kontur des Polygons wird mit der Farbe des aktuellen Pinsels gezeichnet.

BEISPIEL

```
•TabDef(PolLeer,10)
•TabDef(PolVoll,10)
•TabDef(PolOffen,10)
•TabZufall(PolLeer,100,400,2)
•TabZufall(PolVoll,100,400,2)
•TabZufall(PolOffen,100,400,2)
•Füllfarbe(247)
•Pinsel(241)•Polygon(PolLeer)
•Pinsel(242)•Polygon(PolVoll,+)
•Pinsel(243)•Polygon(PolOffen,-)
```

SIEHE AUCH •**Rechteck**(), •**RRechteck**()

Alias: (Pos) Befehl: Pos

Positionieren des Textes

SYNTAX •**Pos(*X,Y*)**

BESCHREIBUNG Positionierung des Textcursors an den Koordinaten ***X,Y***

X,Y : Koordinaten des Textcursors

ANMERKUNG Dieser Befehl bezieht sich auf das aktive Kommentarfenster.

BEISPIEL

```
•Pos(200,150)DER TEXT BEGINNT AN DER POSITION 200,150
```

SIEHE AUCH •**KomFen**()

Alias: (Point) Befehl: Punkt

Zeichnen eines Punktes

SYNTAX •**Punkt** ***(X,Y)***

BESCHREIBUNG Zeichnet einen Punkt an den Koordinaten ***X,Y***

X,Y : Koordinaten des Punktes

ANMERKUNG Der Punkt wird in der Farbe, dem Modus und dem Stil des aktuellen Pinsels gezeichnet.

BEISPIEL

```
•Punkt(320,200)
```

SIEHE AUCH •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Ptc) Befehl: PunktFarbe

Ermitteln der Farbe eines Bildschirmpunktes

SYNTAX •**PunktFarbe(*x,y,Index,R,G,B*)**

BESCHREIBUNG Ermittelt die Farbe eines Bildschirmpunktes mit den Koordinaten ***x,y***.

Die Farbe kann auf zwei Arten ermittelt werden:

- durch Angabe ihres Index in der Farbpalette
- durch Angabe der Komponenten Rot, Grün, Blau

x,y	:	Koordinaten des Punktes
Index	:	Name der Variablen, in der der Index gespeichert wird
R,G,B	:	Namen der Variablen, in denen die Komponenten Rot, Grün, Blau gespeichert werden

BEISPIEL

```
•PunktFarbe(100,200,Index,Rot,Grün,Blau)
```

SIEHE AUCH •**RGBIndex**()

Alias: (Calc) Befehl: Rechner

Aktivieren bzw. desaktivieren der Möglichkeit der Benutzung des Rechners durch den Schüler

SYNTAX •**Rechner(+)** oder •**Rechner(-)**

BESCHREIBUNG Aktiviert bzw. desaktiviert die Möglichkeit für den Schüler, eine numerische Antwort mit Hilfe eines Klammerausdrucks einzugeben.

•***Rechner(+)*** : läßt einen Klammerausdruck zu

•***Rechner(-)*** : verbietet einen Klammerausdruck

Alias: (Rect) Befehl: Rechteck

Zeichnen eines Rechtecks

SYNTAX •**Rechteck(*Xh,Yh,Xb,Yb*)**

BESCHREIBUNG Zeichnet ein Rechteck, das durch die Koordinaten seiner Diagonalen ***Xh,Yh,Xb,Yb*** beschrieben wird

Xh,Yh,Xb,Yb : Koordinaten der Diagonalen des Rechtecks

ANMERKUNG Die Darstellung erfolgt mit der Farbe und im Modus des aktuellen Pinsels.

BEISPIEL

```
•Rechteck(100,150,400,200)
```

SIEHE AUCH •**RRechteck**(), •**Polygon**(), •**Fläche**(), •**Füllen**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (RectLH) Befehl: RechteckBH

Zeichnen eines Rechtecks

SYNTAX •**RechteckBH(*Xh,Yh,B,H*)**

BESCHREIBUNG Zeichnet ein Rechteck, das durch die Koordinaten seiner linken oberen Ecke ***Xh,Yh*** und durch seine Breite ***B*** und seine Höhe ***H*** definiert wird.

Xh,Yh	:	Koordinaten der linken oberen Ecke des Rechtecks
B,H	:	Breite und Höhe des Rechtecks

ANMERKUNG Die Darstellung erfolgt mit der Farbe und dem Modus des aktuellen Pinsels.

BEISPIEL

```
•RechteckBH(100,150,400,200)
```

SIEHE AUCH •**Rechteck**(), •**RRechteck**(), •**Polygon**(), •**Fläche**(), •**Füllen**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (PageOnglet) Befehl: RegSeite

Auswahl einer Seite einer Register-Taste

SYNTAX •**RegSeite(*RegVarName,Seite*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Seite von einer Winkeltaste auszuwählen, auf welcher man alle Arten von Tasten verbinden kann (außer dem Menü).

RegVarName : Name der Variablen, die mit der Registertaste verbunden ist.

Seite : Seitennummer des Registers

ANMERKUNG Nach der Erstellung einer Registertaste mittels des Befehls •**TasteReg**() muß man jede dieser Seiten auswählen, um ihnen die Tasten zuzuordnen.

Um die Erstellung der Tasten auf einer Seite zu beenden, muß man den Befehl •**RegSeite**() ohne Parameter ausführen.

BEISPIEL

```
•TasteReg(TReg,'Register1,Register2,Register3',10,10,400,470)
•RegSeite(TReg,1)
•Taste(O1_1,TASTE auf der Seite 1',100,100,100,50)
•TasteReg(TReg,2)
•Taste(O2_1,TASTE auf der Seite 2',100,100,100,50)
•TasteReg(TReg,3)
•Taste(O3_1,TASTE auf der Seite 3',100,100,100,50)
•TasteReg()
```

SIEHE AUCH •**RegSeite**()

Alias: (InitMod) Befehl: Reset

Rücksetzen von verschiedenen Optionen auf ihren Ursprungswert

SYNTAX

- **Reset()**

BESCHREIBUNG Dieser Befehl läßt folgende weitere Befehle ausführen:

- **Sonderzeichen**(+)
- **GrBuchst**(+)
- **KlBuchst**(-)
- **ZeiSetzung**(+)
- **Sam**(-)
- **ScrollDef**(-)
- **Ausrichten**(0)
- **ZeiMod**(1)
- **PinselStil**(1)
- **KomFen**()
- **AntwFen**()
- **Bsch**(0)
- **ClipDef**()
- **MausZone**()
- **Achsen**(-)

BEISPIEL

```
•Reset()
```

Alias: (RVBI) Befehl: RGBIndex

Ermitteln der RGB-Komponenten des Index einer Farbe

SYNTAX •**RGBIndex(*Index,R,G,B,Pal*)**

BESCHREIBUNG Speichert in den Variablen ***R, G, B*** die Komponenten rot, grün und blau der Farbe mit dem Index ***Index*** in der gewählten Palette.

Index	:	Index der Farbe
R,G,B	:	Namen der numerischen Variablen, in denen die Komponenten rot, grün und blau gespeichert werden
Pal	:	Nummer der gewählten Palette (0 bis 16)

ANMERKUNG Im Normalfall ist die ausgewählte Palette die physische Palette (0)

BEISPIEL

```
•RGBIndex(10,R,G,B)
```

SIEHE AUCH •**RGBWechsel**()

Alias: (IRVB) Befehl: RGBWechsel

Wechseln einer Farbe in der aktuellen Farbpalette

SYNTAX •**RGBWechsel(*Index,R,G,B,Pal*)**

BESCHREIBUNG Wechselt eine Farbe in der aktuellen Palette.

Index	:	Index der zu wechselnden Farbe
R,G,B	:	Komponenten Rot, Grün, Blau
Pal	:	Nummer der gewählten Palette (0 bis 16)

ANMERKUNG Im Normalfall ist die ausgewählte Palette die physische Palette (0)

BEISPIEL

```
•RGBWechsel(15,120,50,255,2)
```

SIEHE AUCH •**Farbskala**(), •**RGBIndex**()

Alias: (RRect) Befehl: RRechteck

Zeichnen eines Rechtecks mit abgerundeten Ecken

SYNTAX •**RRechteck(*Xh,Yh,Xb,Yb,Rx,Ry,+*)**

BESCHREIBUNG Zeichnet ein Rechteck, das durch die Koordinaten ***Xh,Yh,Xb,Yb*** seiner Diagonalen definiert wird

Xh,Yh,Xb,Yb	:	Koordinaten der Diagonalen des Rechtecks
Rx,Ry	:	Rundungs-Radien
+	:	Option für das Füllen des Rechtecks

ANMERKUNG Die Darstellung erfolgt mit der Farbe und dem Modus des aktuellen Pinsels.

Standardmäßig ist der Radius proportional zur Breite und Länge des Rechtecks.

BEISPIEL

```
•RRechteck(100,150,400,200,10,10)
•RRechteck(100,150,400,200,,,+)
```

SIEHE AUCH •**Rechteck**(), •**Polygon**(), •**Fläche**(), •**Füllen**(), •**ZeiMod**(), •**Pinsel**(), •**PinselStil**()

Alias: (RRectLH) Befehl: RRechteckBH

Zeichnen eines Rechtecks mit abgerundeten Ecken

SYNTAX •**RRechteckBH(*Xh,Yh,B,H,Rx,Ry,+*)**

BESCHREIBUNG Zeichnet ein Rechteck, das durch die Koordinaten ***Xh,Yh*** der linken oberen Ecke des Rechteckes und der Breite ***B*** und der Höhe ***H*** definiert wird

Xh,Yh	:	Koordinaten der linken oberen Ecke des Rechtecks
B,H	:	Breite und Höhe des Rechteckes
Rx,Ry	:	Rundungs-Radien
+	:	Option für das Füllen des Rechtecks

ANMERKUNG Die Darstellung erfolgt mit der Farbe und dem Modus des aktuellen Pinsels. Standardmäßig ist der Radius proportional zur Breite und Länge des Rechtecks.

BEISPIEL

```
•RRechteckBH(100,150,50,80,10,10)
•RRechteckBH(100,150,50,80,,,+)
```

SIEHE AUCH •**Rechteck**(), •**Polygon**(), •**Fläche**(), •**Füllen**(), •**ZeiMod**(), •**Pinsel**(), •**PinselStil**()

Alias: (Ram) Befehl: Sam

Festlegen eines Zahleneinsammlers

SYNTAX •**Sam(*VarName*)** oder •**Sam(*VarName,Anzahl*)** oder •**Sam(-)**

BESCHREIBUNG Initialisiert den Zahleneinsammler. Alle in der Mitteilung des Schülers aufscheinenden Zahlen werden eingesammelt und automatisch von der numerischen Variable ***VarName*** an in die Liste der Variablen gespeichert, die mit dem Befehl •**SamList**() definiert wurde.

- •***Sam(VarName)*** : Sammelt alle Zahlen ein, die in der Mitteilung des Schülers vorkommen. Die Anzahl der eingesammelten Zahlen wird nur durch die Anzahl der Variablen limitiert, die mit dem Befehl •**SamList**() definiert wurde.
- •***Sam(VarName,Anzahl)*** : Sammelt alle Zahlen ein, die in der Antwort des Schülers vorkommen, wobei die ***Anzahl*** auf Anzahl Zahlen limitiert ist. Wenn ***Anzahl*** größer als die Anzahl der Variablen in der Liste ist, so werden höhere Anzahl ignoriert und das Einsammeln wird bei der letzten Variablen beendet.
- •***Sam(-)*** : Desaktiviert den Zahleneinsammler

ANMERKUNG Wenn keine Variablenliste definiert wurde, wird nur die erste Zahl eingesammelt. Achtung: Der Zahleneinsammler bleibt solange aktiv, bis er mit dem Befehl •**Sam**(-) desaktiviert wird.

BEISPIEL

```
•Sam(Blau,3)
```

SIEHE AUCH •**SamList**(), •**ChaSam**()

Alias: (RamList) Befehl: SamList

Erzeugen einer Liste für das Zahleneinsammeln

SYNTAX •**SamList(*VarName_1,..,VarName_N*)**

BESCHREIBUNG Erzeugt eine Liste mit den Namen der Variablen, die für das Zahleneinsammeln verwendet wird

VarName_1,,VarName_N : Name der numerischen Variablen, die die Liste für das Zahleneinsammeln darstellt

BEISPIEL

•SamList(Rot,Blau,Gelb,Weiß)

SIEHE AUCH •**Sam**()

Alias: (Remonte) Befehl: Schleife

Definieren der Rückkehr zu einer Zeile

SYNTAX

- **Schleife(*Anzahl,Zeilennummer*)** oder
- **Schleife(*Anzahl,Label*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Definition einer Schleife in einem Text oder im Kommentarteil. Sie bedeutet: ***Anzahl*** - mal zur durch eine Nummer oder einen Label angegebenen Zeile zurückspringen.

Anzahl : Anzahl an Schleifen durchlaufen

Zeilennummer oder

Label : Nummer oder Label der Zeile

ANMERKUNG Die Nummer oder der Label der Zeile müssen in der Zeile, die dem Befehl •**Schleife**() vorangeht, angegeben werden und nicht danach. Wenn die Nummer der Zeile oder der Label durch das Zeichen = (Gleichheitszeichen) ersetzt wird, so erfolgt der Rücksprung zu dieser aktuellen Zeile. Das Zeichen = kann von einer negativen Zahl von relativen Zeilen im Bezug zur laufenden Zeile gefolgt sein. Zum Beispiel springt •**Schleife(10,=-3)** 10 mal auf die laufende Zeile minus drei.

BEISPIEL

```
          [x:=100]
HIER      •Rechteck([x],100,[x+50],200) [x:=x+50]
          ••kehrt 10 mal zur Zeile 2 zurück
          •Schleife(10,HIER)
```

SIEHE AUCH •**Sprung**(), •**SprungBlock**()

Alias: (Defil) Befehl: ScrollDef

Ermöglichen des Scrollens (vertikales Weiterbewegen) in einem Kommentarfenster

SYNTAX •**ScrollDef(+)** oder •**ScrollDef(-)**

BESCHREIBUNG Ermöglicht das Scrollen (vertikales Weiterbewegen) in einem Textfenster. Wird nichts anderes angegeben, so ist ein vertikales Weiterbewegen möglich.

- •***ScrollDef(+)*** : ermöglicht das Scrollen
- •***ScrollDef(-)*** : untersagt das Scrollen

BEISPIEL

```
•ScrollDef(+)
•KomFen(100,100,200,50,243,236)
[x:=1]
••Scrollen....(Zeile [x])
[x:=1] •Schleife(10,-1)
```

SIEHE AUCH •**KomFen**()

Alias: (HVDecal) Befehl: ScrollText

Horizontales oder vertikales Scrollen eines Textes in einem Textfenster

SYNTAX •**ScrollText(*Richtung,Nb,Div*)**

BESCHREIBUNG Bewirkt eine horizontale oder vertikale Bewegung eines Textes im aktiven Textfenster.

Richtung	:	Richtung der Fortbewegung 0 = nach unten 1 = nach oben (ohne Angabe) 2 = nach links 3 = nach rechts
Nb	:	Anzahl an Zeilen, die auf einmal übersprungen werden sollen (Ohne Angabe bedeutet: Sprung um 1 Zeile)
Div	:	Divisor der Höhe einer Zeile (ohne Angabe = 1) (Ohne Angabe bedeutet: Höhe der Zeile ist 1)

ANMERKUNG Der Wert der Fortbewegung ist eine Funktion der aktuellen Schrift, sowohl für die vertikale als auch für die horizontale Fortbewegung. Dieser Wert wird durch den Parameter ***Div*** dividiert. Der Parameter ***Nb*** gibt die Anzahl der Fortbewegungen an.

BEISPIEL

```
                •Aktion(Ende)
                •KomFen(100,100,30,150,236,239)

rechts          •ScrollText(3,2,15)      ••nach rechts
                •Schleife(60,Rechts)
unten           •ScrollText(0,1,8)       ••nach unten
                •Schleife(60,Unten)
links           •ScrollText(2,2,15)      ••nach links
                •Schleife(60,links)
oben            •ScrollText(1,1,8)       ••nach oben
                •Schleife(60,oben)
                •Sprung(rechts)
                •Ende()
```

SIEHE AUCH •**KomFen**(), •**ScrollDef**()

Alias: (Diacrit) Befehl: Sonderzeichen

Option um verschiedene Sonderzeichen (z.B. é, ü ...) in die äquivalenten Zeichen (e, u ...) umzuwandeln

SYNTAX

- **Sonderzeichen(+)** oder
- **Sonderzeichen(-)** oder
- **Sonderzeichen(*Quelle,Ziel*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Definition einer Option, die bei der Editierung der Schülereingabe Sonderzeichen in die äquivalenten Zeichen konvertiert, oder Sonderzeichen in einer Zeichenkette in die äquivalenten Zeichen umwandelt.

- ***Sonderzeichen(+)*** : aktiviert die Konvertierung
- ***Sonderzeichen(-)*** : desaktiviert die Konvertierung
- ***Sonderzeichen(Quelle,Ziel)*** : Umwandlung von Quelle in Ziel ohne Sonderzeichen

BEISPIEL

```
•Sonderzeichen(Mitteilung,OhneSonderz)
```

SIEHE AUCH •**GrBuchst**(), •**KlBuchst**(), •**ZeiSetzung**()

Alias: (Saut) Befehl: Sprung

Springen zu einer Zeile

SYNTAX •**Sprung(*Zeilennummer*)** oder •**Sprung(*Label*)**

BESCHREIBUNG Dieser Befehl ermöglicht den Sprung zu einer Text- oder Kommentarzeile nach oben oder unten.

Zeilennummer oder ***Label*** : Nummer oder Label der Zeile

ANMERKUNG Wenn die ***Zeilennummer*** oder der ***Label*** durch das Zeichen = (Gleichheitszeichen) ersetzt wird, so ist die aktuelle Zeile das Ziel des Sprunges. Das Zeichen = kann von einer negativen oder positiven Zahl von relativen Zeilen im Bezug zur laufenden Zeile gefolgt sein. Zum Beispiel springt •**Sprung(=+5)** 5 Zeilen unter die laufende Zeile.

BEISPIEL

```
•Sprung(Ausstieg)
•Sprung(=-3)
```

SIEHE AUCH •**SprungBlock**(), •**Schleife**()

Alias: (SautBloc) Befehl: SprungBlock

Springen zu einem Analyseblock

SYNTAX •**SprungBlock(*Blocknummer*)** oder •**SprungBlock(*Label*)**

BESCHREIBUNG Dieser Befehl ermöglicht den Sprung zu einem Analyseblock, ausgehend von einem Text oder Kommentar.

Blocknummer oder ***Label*** : Nummer oder Label des Blockes

ANMERKUNG Der Block 0 ist der Textteil des Moduls. Wenn die Zeilennummer oder der Label durch das Zeichen = (Gleichheitszeichen) ersetzt wird, so ist das Ziel des Sprunges der aktuelle Block. Das Zeichen = kann von einer negativen oder positiven Zahl von relativen Blöcken im Bezug zum laufenden Block gefolgt sein. Zum Beispiel springt •**SprungBlock(=+5)** 5 Blöcke unter den laufenden Block.

BEISPIEL

```
•SprungBlock(10)
```

SIEHE AUCH •**Sprung**(), •**Schleife**()

Alias: (Stdpal) Befehl: StandPal

Aktivierung/Desaktivierung der Standard-Palette

SYNTAX •**StandPal(+/-)**

BESCHREIBUNG Normalerweise verwendet EGO eine Palette, wo die 20 System-Farben am Ende der Palette, von Index 236 bis 255 angesiedelt sind. Die Standard-Palette von Microsoft ordnet die 20 Systemfarben am Anfang und am Ende der Palette an, von Index 0 bis 9 und 245 bis 255.

- •***StandPal(+)*** : Aktiviert den Standard Paletten-Modus
- •***StandPal(-)*** : Desaktiviert den Standard Paletten-Modus (Standardeinstellung)

BEISPIEL

```
•StandPal(+)
```

Alias: (RFI) Befehl: SucheZKDatei

Suche in Zeichenketten-Dateien

SYNTAX •**SucheZkDatei(*Wort,ZKDat{B},ZKDat{E},Res,Halt,Modus,WortNr, Trennz*)**

BESCHREIBUNG Dieser Befehl sucht in den Zeichenketten ***ZKDat{A}*** bis **ZKDat{E}** einer Zeichenketten-Datei das Vorhandensein eines Wortes. Wenn dieser Befehl ausgeführt ist, beinhaltet die Variable ***Res*** die Liste mit den Nummern der Zeichenketten, die dads Wort ***Wort*** enthalten. Die Liste ***Res*** hat das Leerzeichen als Trennzeichen.

Wort	:	Mane der Zeichenkette, die das gesuchte Wort oder den gesuchten Ausdruck enthält
ZKDat{A}	:	Name der Zeichenkette einer Datei, ab der die Suche begonnen werden soll
ZKDat[E}	:	Name der Zeichenkette einer Datei, bis zu der gesucht werden soll
Res	:	Name der Zeichenkette, welche das Resultat der Suche enthält
Halt	:	Name der numerischen Variablen, die die Nummer der Zeichenkette enthält, wo die Suche beendet wurde
Modus	:	Art der Suche: 0 = Wort nicht vorhanden - Groß/Kleinschreibung inaktiv 1 = Wort vorhanden(Standardeinstellung) - Groß/Kleinschreibung inaktiv 10 = Wort nicht vorhanden - Groß/Kleinschreibung aktiv 11 = Wort vorhanden - Groß/Kleinschreibung aktiv
WortNr	:	Nummer des Wortes der Zeichenkettedatei, in dem die Suche durchgeführt wird (optional)
Trennz	:	Trennzeichen zur Unterteilung der Zeichenkette (optional)

ANMERKUNG Wenn die Variable ***Halt*** einen negativen Wert enthält, dann ist die Zeichenkette ***Res*** voll. Man muß danach eine neuerliche ***SucheZKDatei***, ausgehend von Absolutwert von ***Halt*** durchführen.

BEISPIEL

```
[Wort:='EGO Win']
•SucheZkDatei(Wort,System{1},System{100},Resu,ltat,Halt)
```

SIEHE AUCH •**XFi**(),

Alias: (Syst) Befehl: System

Verwaltung von Verzeichnissen (Directory)

SYNTAX

- **System(*md Pfad*)** oder
- **System(*cd Pfad*)** oder
- **System(*rd Pfad*)**

BESCHREIBUNG Erstellt, wählt oder löscht ein Verzeichnis.

md Pfad	:	Erstellen eines Verzeichnisses
cd Pfad	:	Wechseln eines Verzeichnisses
rd Pfad	:	Löschen eines Verzeichnisses

ANMERKUNG Der Weg ist in der Standardform, ohne \ am Ende, zu schreiben.

BEISPIEL

```
•System(md d:\Briefe)
•System(cd d:\Briefe)
```

SIEHE AUCH •**DateiKopie**(), •**DateiLösch**(), •**DateiUmben**()

Alias: (TabCha) Befehl: TabCha

Kopieren einer Zahlentabelle in eine Zeichenkettendatei

SYNTAX •**TabCha(*Tabellenname,Dateiname,Anzahl*)**

BESCHREIBUNG Kopiert die Zahlen einer Tabelle in eine Zeichenkettendatei.

Tabellenname	:	Name der Zahlentabelle
Dateiname	:	Name der Zeichenkettendatei
Anzahl	:	Anzahl der Zahlen pro Kettenvariablen

ANMERKUNG Die Zeichenkettendatei enthält so viele Kettenvariablen, wie für das Speichern der gesamten Tabelle nötig ist. Die Zahlen der Tabelle werden mit der Anzahl von Nachkommastellen gespeichert, die mit dem Befehl •**Komma**() festgelegt wurden.
Der optionale Parameter ***Anzahl*** gibt die Anzahl von Zahlen pro Kette an.
Standardmäßig wird unter Berücksichtigung der gewünschten Präzisierung die maximal mögliche Anzahl an Zahlen (max. 1024 Zeichen) in eine Kettenvariable geschrieben.

BEISPIEL

```
•TabDef(Tabelle,20,30)
•TabZufall(Tabelle,1,3,1)
•Komma(8)
•TabCha(Tabelle,TabSpeich,10)
•ChaTab(TabSpeich,Tabelle)
```

SIEHE AUCH •**ChaTab**(), •**VarCha**(), •**VarLösch**(), •**Komma**()

Alias: (DimTab) Befehl: TabDef

Definieren und dimensionieren eine Zahlentabelle

SYNTAX •**TabDef(*Tabellenname,d1[,d2,d3*])**

BESCHREIBUNG Definiert und dimensioniert eine Zahlentabelle

Tabellenname : Name der Tabelle

d1,d2,d3 : Dimensionen der Tabelle

ANMERKUNG Eine Tabelle kann ein-, zwei- oder dreidimensional definiert werden. Die maximale Anzahl an Tabellenelementen ist 8000.

Eine bereits dimensionierte Tabelle kann ohne Verlust ihrer Elemente neu dimensioniert werden.

BEISPIEL

```
•TabDef(Tabelle,10,10,1)
```

SIEHE AUCH •**TabFüllen**(), •**TabZufall**(), •**TabDim**()

Alias: (TabDim) Befehl: TabDim

Ermitteln der Dimensionen einer Zahlentabelle

SYNTAX •**TabDim(*Tabellenname,Zeile,Spalte,Tiefe*)**

BESCHREIBUNG Ermittelt die Dimensionen einer Zahlentabelle und speichert sie in den numerischen Variablen ***Zeile***, ***Spalte***, ***Tiefe***.

Tabellenname	:	Name der Tabelle
Zeile	:	Name der numerischen Variable, in der die Anzahl der Zeilen der Tabelle gespeichert wird
Spalte	:	Name der numerischen Variable, in der die Anzahl der Spalten der Tabelle gespeichert wird
Tiefe	:	Name der numerischen Variable, in der die Anzahl der Elemente der dritten Dimension gespeichert wird

ANMERKUNG Wenn eine der numerischen Variablen den Wert Null hat, so existiert die Dimension, die sie repräsentiert, nicht.

BEISPIEL

```
•TabDef(TAB,10,10,20)
•TabDim(TAB, L, C, P)
[L]/[C]/[P]                    ••zeigt an: 10/10/20
```

SIEHE AUCH •**TabDef**()

Alias: (TabR) Befehl: TabFüllen

Füllen einer Tabelle

SYNTAX

•**TabFüllen(*Tabellenname,Wert*)** oder

•**TabFüllen(*Tabellenname,Orig,Inkrement*)**

BESCHREIBUNG Füllt eine Tabelle ***Tabellenname*** mit Wert ***Wert*** aus oder füllt die Tabelle ***Tabellenname*** mit einer arithmetischen Progression, die vom Wert ***Orig*** ausgeht und dem Veränderungswert ***Inkrement*** hat.

Tabellenname	:	Name der Zahlentabelle
Wert	:	Füllwert
Orig	:	Ausgangswert der arithmetischen Progression
Inkrement	:	Veränderungswert (Inkrement)

ANMERKUNG ***Wert***, ***Orig*** und ***Inkrement*** sind ganze Zahlen oder Dezimalzahlen.

BEISPIEL

```
•Tabdef(Tab1,10,10)
•TabFüllen(Tab1,0)      ••Füllt Tab1 mit Nullen
•DimTab(Tab2,50)
•TabFüllen(Tab2,5,2)    ••Füllt Tab2 mit dem Ausgangswert 5
                          ••und dem Veränderungswert 2
```

SIEHE AUCH •**TabZufall**(), •**TabDef**()

Alias: (TabCop) Befehl: TabKopie

Kopieren einer Tabelle in eine andere Tabelle

SYNTAX •**TabKopie(*NameQuelltab,NameZieltab*)**

BESCHREIBUNG Kopiert den Inhalt der Zahlentabelle ***NameQuelltab*** in die Zahlentabelle ***NameZieltab***.

NameQuelltab : Name der Quelltabelle

NameZieltab : Name der Zieltabelle

ANMERKUNG Die Anzahl der kopierten Elemente entspricht der Zahl der Elemente der kleineren der beiden Tabellen.

BEISPIEL

```
•TabDef(Tab1,10,10)
•TabFüllen(Tab1, 1,1)
•DimTab(Tab2,50)
•TabKopie(Tab1,Tab2)
```

SIEHE AUCH •**TabFüllen**(), •**TabZufall**(), •**TabDef**()

Alias: (TabTri) Befehl: TabOrd

Ordnen der Werte einer Tabellen

SYNTAX •**TabOrd(*Tabellenname,Richtung*)**

BESCHREIBUNG Ordnet die Werte einer Tabelle aufsteigend oder absteigend, je nach definierter (vorgegebener) Richtung.

Tabellenname	:	Name der Zahlentabelle
Richtung	:	Art der Anordnung: 0 = aufsteigend 1 = absteigend (Standardeinstellung = 0)

BEISPIEL

```
•TabDef(TAB1,10,10)
•TabZufall(TAB1,200)
•TabOrd(Tab1)                    ••aufsteigende Anordnung
```

SIEHE AUCH •**TabZufall**(), •**TabFüllen**(), •**TabDef**()

Alias: (TabRH) Befehl: TabZufall

Füllen einer Tabelle mit Zufallszahlen

SYNTAX •**TabZufall(*Tabellenname,A,B,Typ*)**

BESCHREIBUNG Füllt die Tabelle ***Tabellenname*** mit Zufallszahlen zwischen ***A*** und ***B***. Der Parameter ***Typ*** definiert die Art der Ziehung.

Tabellenname	:	Name der Zahlentabelle
A	:	Unterer Grenzwert der Zufallsziehung
B	:	Oberer Grenzwert der Zufallsziehung

Typ	**Art der Ziehung**
0	= Zufallsziehung von ganzen Zahlen zwischen A und B ohne Kontrolle von doppelt vorkommenden Zahlen (Standardeinstellung)
1	= Zufallsziehung von Dezimalzahlen zwischen A und B ohne Kontrolle von doppelt vorkommenden Zahlen
2	= Zufallsziehung von verschiedenen ganzen Zahlen zwischen A und B
3	= Zufallsziehung von verschiedenen Dezimalzahlen zwischen A und B

ANMERKUNG ***A*** und ***B*** sind ganze Zahlen oder Dezimalzahlen. Der obere Grenzwert ***B*** ist von der Zufallsziehung ausgeschlossen.

Im Falle einer Zufallsziehung vom Typ 2 muß der Intervall zwischen A und B größer sein als die Dimensionen der zu füllenden Tabelle, sonst führt der Befehl eine Zufallsziehung des Typs 0 durch.

BEISPIEL

```
•TabDef(TAB1,10,10)
•TabZufall(TAB1,0,200)••Füllt TAB1 mit Zufallszahlen
                       ••zwischen 0 und 199
•TabDef(TAB2,50)
•TabFüllen(TAB2,0,1.5)••Füllt TAB2 mit dezimalen
                       ••Zufallszahlen zwischen
                       ••0 und 1.5 (1,5 ausgeschlossen)
```

SIEHE AUCH •**TabFüllen**(), •**TabDef**()

Alias: (Clavier) Befehl: Tastatur

Aktiviert / desaktiviert die Tastatur oder simuliert einen Anschlag auf ein Zeichen

SYNTAX

•**Tastatur(+)** oder

•**Tastatur(-)** oder

•**Tastatur(*Text*)**

BESCHREIBUNG Dieser Befehl erlaubt das aktivieren bzw. desaktivieren der Tastatur oder das Simulieren eines Anschlags auf ein Zeichen oder eine Funktionstaste

•***Tastatur(+)*** : Aktiviert die Tastatur (Grundeinstellung)

•***Tastatur(-)*** : Desaktiviert die Tastatur

•***Tastatur(Text)*** : Simuliert das Anschlagen von Zeichen

ANMERKUNG Wenn die Tastatur desaktiviert ist, so werden alle auf ihr stattfindenden Aktionen ignoriert, nur die Maus bleibt aktiv. Um den Anschlag von Zeichen zu simulieren, muß man sie wie Parameter eines Befehls behandeln. Um Funktionstasten wie **F1, Zeilenschaltung** oder auch **ALT F** zu simulieren, muß man ihrem Code ein °**F** vorangehen lassen. Um den Anschlag von Zeichen, ausgehend von ihrem ASCII-Code zu simulieren, muß man ihrem Code ein °**S** vorangehen lassen.

BEISPIEL

```
•Extern(write,11)            ••startet Write
••AktivExtern(write)
••einen Text in Write schreiben
•Tastatur('Text °F13in Write°F13in mehreren Zeilen')
•Pause(5)
••speichern des Dokumentes
•Tastatur('°F470e')          ••Alt F e
•Pause(5)
••dem Dokument einen Namen geben
•Tastatur('Demo')•Pause(3)•Tastatur('°F13')
••Write verlassen
•Tastatur('°F470q')
```

SIEHE AUCH •**AktivExtern**(), •**MausKlick**(),

Alias: (Bouton) Befehl: Taste

Definieren einer Windows-Taste

SYNTAX •**Taste(*VarName,Text,Xo,Yo,Breite,Höhe*)**

BESCHREIBUNG Definiert und erstellt eine Windows-Taste, deren obere linke Ecke die Koordinaten ***Xh,Yh,*** die Breite ***Breite*** und die Höhe ***Höhe*** hat.

VarName	:	Name einer numerischen globalen oder lokalen Variable, die mit der Taste verbunden wird
Text	:	Text, der in der Taste erscheint
Xo,Yo	:	Koordinaten der oberen linken Ecke der Taste
Breite	:	Breite der Taste in Pixel
Höhe	:	Höhe der Taste in Pixel

ANMERKUNG Sobald die Taste angeklickt wird, wird der zugeordneten Variablen ***VarName*** der Wert 1 zugewiesen. Bei der Rückkehr zum Interventionspunkt des Schülers nimmt die Variable wieder den Wert 0 an.

Die zugeordnete Variable kann eine globale oder lokale sein. Wenn es sich um eine lokale Variable handelt, so ist auch die Taste lokal an das Modul, das es erzeugt, gebunden und verschwindet bei der Verkettung mit einem anderen Modul.
Wenn im Text ein Beistrich oder ein Apostroph vorkommt, so muß er zwischen Apostrophe geschrieben werden. Man kann 600 Tasten zugleich erzeugen.

Der Befehl •**TasteLösch**() ermöglicht die Unterdrückung der Taste.

BEISPIEL

```
•Taste(VHilfe,Hilfe,500,10,50,40)
```

SIEHE AUCH •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**(),

Alias: (BtnCoche) Befehl: TasteAuswahl

Definieren einer Auswahl-Taste

SYNTAX •**TasteAuswahl(*VarName,Text,Xo,Yo,Breite,Höhe*)**

BESCHREIBUNG Definiert und erstellt eine Auswahl-Taste, deren linke obere Ecke die Koordinaten ***Xo,Yo***, die Breite ***Breite*** und die Höhe ***Höhe*** hat.

VarName	:	Name einer lokalen oder globalen numerischen Variable, die mit der Taste verbunden ist
Text	:	Text, der in der Taste erscheint
Xo,Yo	:	Koordinaten der oberen linken Ecke der Taste
Breite	:	Breite der Taste in Pixel
Höhe	:	Höhe der Taste in Pixel

ANMERKUNG Sobald die Taste angeklickt wird, wird der zugeordneten Variablen ***VarName*** der Wert 1 zugewiesen, sonst ist der Wert 0. Die zugeordnete Variable kann eine globale oder lokale sein. Wenn es sich um eine lokale Variable handelt, so ist auch die Taste lokal an das Modul, das es erzeugt gebunden und verschwindet bei der Verkettung mit einem anderen Modul.

Wenn im Text ein Beistrich oder ein Apostroph vorkommt, muß er zwischen Apostrophe geschrieben werden. Man kann 600 Dialogfelder (Tasten) zugleich erzeugen.

Der Befehl •**TasteLösch**() ermöglicht das Löschen der Taste.

BEISPIEL

```
•Pinsel(236)•Füllfarbe(255)
•TasteAuswahl(Opt1,Option 1,500,10,50,40)
•TasteAuswahl(Opt2,Option 2,500,10,50,40)
•TasteAuswahl(Opt3,Option 3,500,10,50,40)
•TasteModus(Opt1,2)     ••Auswählen der ersten Taste
```

SIEHE AUCH •**Taste**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnCombo) Befehl: TasteCombo

Definition eines Fensters vom Typ COMBOBOX (statisch)

SYNTAX •**TasteCombo(*VarName,ZDat{A},ZDat{E},X,Y,B,H,Trennz,WortNr*)**

BESCHREIBUNG Definiert und erstellt ein Fenster, mit den Koordinaten der linken oberen Ecke ***X,Y,*** der Breite **B**, der Höhe ***H***, mit Bildlaufleisten, mit denen sich die Zeichenketten der Zeichenketten-Datei ***ZDat*** anzeigen lassen.

VarName	:	Name der lokalen oder globalen Variablen, die mit der Liste verbunden ist
ZDat{A}	:	Erste Zeichenkette der Zeichenkettendatei
ZDat{E}	:	Letzte Zeichenkette der Zeichenkettendatei
X,Y	:	Koordinaten der linken oberen Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Trennz	:	Mögliches Trennzeichen zur Teilung der ZKette
WortNr	:	Nummer des Wortes der Zeichenkette, welches im Fenster erscheint

ANMERKUNG Wenn ***WortNr*** gesetzt wurde, dann werden im Fenster die Worte mit der Nummer WortNr der Zeichenkettendatei ***ZDat*** präsentiert, wobei als Trennzeichen ***Trennz*** dient. Die Variable, die mit VarName verbunden ist, beinhaltet 2 Wörter, mit dem senkrechten Strich (I) als Trennzeichen. Das erste Wort beinhaltet die Nummer des Items, auf das man geklickt hat, das zweite Wort enthält den Namen des Items, auf das man geklickt hat. Die Systemvariable ***ewControle*** beinhaltet den Namen der Variablen, bei dem ein Item mit einem Doppelklick durchgeführt wurde.

Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist Fenster lokal an das Modul gebunden, durch das es erzeugt wurde und verschwindet anläßlich der Verkettung mit einem anderen Modul. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden.

Der Befehl •**TasteLösch**() ermöglicht das Löschen des Fensters.

BEISPIEL

```
•dir(c:\windows\*.ini,dir)
•TasteCombo(DirListe,dir{1},,200,200,150,200)
•Pause()
•Eingabe()
[DirListe]
•TasteCombo(NameListe,Name{1},,350,200,100,200,'/',2)
•Eingabe()
[NameListe]
•Pause()
•Ende()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnComboEdit) Befehl: TasteComboEdit

Definition eines Fensters vom Typ COMBOBOX mit Editor

SYNTAX •**TasteComboEdit(*VarName,ZDat{A},ZDat{E},X,Y,B,H,Trennz,WortNr*)**

BESCHREIBUNG Definiert und erstellt ein Fenster, mit den Koordinaten der linken oberen Ecke ***X,Y,*** der Breite ***B***, der Höhe ***H*** und mit Bildlaufleisten, mit denen sich die Zeichenketten der Zeichenketten-Datei ***ZDat*** anzeigen lassen.

VarName	:	Name der lokalen oder globalen Variablen, die mit der Liste verbunden ist
ZDat{A}	:	Erste Zeichenkette der Zeichenkettendatei
ZDat{E}	:	Letzte Zeichenkette der Zeichenkettendatei
X,Y	:	Koordinaten der linken oberen Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Trennz	:	Mögliches Trennzeichen zur Teilung der ZKette
WortNr	:	Nummer des Wortes der Zeichenkette, welches im Fenster erscheint

ANMERKUNG Wenn ***WortNr*** gesetzt wurde, dann werden im Fenster die Worte mit der Nummer ***WortNr*** der Zeichenkettendatei ***ZDat*** präsentiert, wobei als Trennzeichen ***Trennz*** dient. Die Variable, die mit ***VarName*** verbunden ist, beinhaltet 2 Wörter, mit dem senkrechten Strich (I) als Trennzeichen. Das erste Wort beinhaltet die Nummer des Items, auf das man geklickt hat, das zweite Wort enthält den Namen des Items, auf das man geklickt hat. Die Systemvariable ***ewControle*** beinhaltet den Namen der Variablen, bei dem ein Item mit einem Doppelklick durchgeführt wurde.

Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist Fenster lokal an das Modul gebunden, durch das es erzeugt wurde und verschwindet anläßlich der Verkettung mit einem anderen Modul. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden.

Der Befehl •**TasteLösch**() ermöglicht das Löschen des Fensters.

BEISPIEL

```
•dir(c:\windows\*.ini,dir)
•TasteComboEdit(DirListe,dir{1},,200,200,150,200)
•Pause()
•Eingabe()
[DirListe]
•TasteComboEdit(NameListe,Name{1},,350,200,100,200,'/',2)
•Eingabe()
[NameListe]
•Pause()
•Ende()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnComboStatic) Befehl: TasteComboFix

Definieren eines Fenster vom Typ COMBOBOX mit immer geöffnetem Editor

SYNTAX •**TasteComboFix(*VarName,ZDat{A},ZDat{E},X,Y,B,H,Trennz,WortNr*)**

BESCHREIBUNG Definiert und erstellt ein Fenster, mit den Koordinaten der linken oberen Ecke ***X,Y***, der Breite ***B***, der Höhe ***H***, mit Bildlaufleisten, in denen sich die Zeichenketten der Datei ***ZDat*** anzeigen lassen.

VarName	:	Namen von einer globalen oder lokalen Variablen-Kette,die an die Liste angeschlossen wird
ZDat{A}	:	Erste Zeichenkette der Liste
ZDat{E}	:	Letzte Zeichenkette der Liste
X,Y	:	Koordinaten der linken oberen Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Trennz	:	Möglicher Seperator zur Teilung der ZKette
WortNr	:	Nummer des Wortes der Zahlenkette, welche im Fenster angezeigt wird

ANMERKUNG Wenn ***WortNr*** gesetzt wurde, dann werden in der Fensterliste die Worte ***WortNr*** der Zeichenkettedatei ***ZDat*** angezeigt, wobei als Seperator ***Trennz*** dient. Die zugeordnete Variablenkette **VarName** beinhaltet 2 Worte mit einem vertikalen Strich (|) für das Trennzeichen. Das erste Wort beinhaltet die Nummer des Items, auf das man geklickt hat und das zweite Wort enthält das Item, auf das man geklickt hat. Die Systemvariable ***ewControle*** beinhaltet den Namen der zugeordneten Variable, sobald bei dem Item ein Doppelklick durchgeführt wurde. Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist das Fenster lokal an das Modul gebunden, durch das es erzeugt wurde und verschwindet bei der Verkettung mit einem anderem Modul. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden.

Der Befehl •**TasteLösch**() ermöglicht das Löschen eines COMBOBOX Fensters.

BEISPIEL

```
•dir(c:\windows\*.ini,dir)
•TasteComboFix(DirListe,dir{1},,200,200,150,200)
•Eingabe()
•TasteComboFix(ListenName,Name{1},,350,200,100,200,'/',2)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnCursor) Befehl: TasteCursor

Definieren einer linearen Cursor-Taste (Bildlaufleiste)

SYNTAX •**TasteCursor(*VarName,Typ,Xo,Yo,B,H,Max,Min,Pos*)**

BESCHREIBUNG Definieren einer linearen horizontalen oder vertikalen Cursor-Taste

VarName	:	Name einer globalen oder lokalen Variable, die mit der Taste verbunden wird
Typ	:	Ausrichtung der Bildlaufleiste 1 = Horizontale Bildlaufleiste; 2 = Vertikale Bildlaufleiste
Xo,Yo	:	Koordinaten der linken oberen Ecke
B	:	Breite der Bildlaufleiste
H	:	Höhe der Bildlaufleiste
Max	:	Maximaler Wert der Extremposition des Cursors. (100 bei Nichtangabe und 32536 maximal)
Min	:	Minimaler Wert der Anfangsposition (0 bei Nichtangabe und 32536 maximal)
Pos	:	Cursor-Position im Verhältnis zur Abstufung 1 = unten für eine horizontale Taste 2 = oben für eine horizontale Taste 3 = links für eine vertikale Taste 4 = rechts für eine vertikale Taste

ANMERKUNG Jede Versetzung des Cursors verringert oder erhöht die zugeordnete Variable ***VarName***. Die zugeordnete Variable kann eine lokale oder globale sein. Wenn es sich um eine lokale Variable handelt, so ist auch die Bildlaufleiste lokal an das Modul, das es erzeugt, gebunden und verschwindet bei der Verkettung mit einem anderen Modul. Im Falle von Negativ-Werten von ***Min*** und ***Max*** erscheint der Cursor in Gegenrichtuing zur Anfangsposition. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden. Der Befehl •**TasteLösch**() ermöglicht das Löschen einer Bildlaufleiste.

BEISPIEL

```
•TasteCursor(Horiz,1,500,200,150,20,,,2)
•TasteCursor(Verti,2,700,10,20,180,50)
•TasteCursor(TiefHoch,2,200,100,20,200,1,150,1)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnEdit) Befehl: TasteEdit

Definieren einer Editier-Fenster-Taste

SYNTAX `TasteEdit(VarName,Text,Xo,Yo,B,H,Max,Erf,Grkl,Versch,PWort)`

BESCHREIBUNG Definiert und erstellt ein Editier-Fenster mit den Koordinaten der linken oberen Ecke ***Xo,Yo***, der Breite ***B*** und der Höhe ***H***.

VarName	:	Name von einer globalen oder lokalen Variablen, die mit der Taste verbunden wird.
Text	:	Text, der im Fenster erscheint
Xo,Yo	:	Koordinaten der linken oberen Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Max	:	Maximale Anzahl der Zeichen, die in einem Fenster erfaßt werden können (1020 ohne Angabe)
Erf	:	Bestimmende Parameter für die Daten-Erfassung 0 = Modus der Eingabe 1 = Modus der Ersetzung 2 = Modus Flip/Flop Eingabe-Ausgabe
GrKl	:	Bestimmende Parameter für die Art der Anzeige der erfaßtenZeichen 0 = wie erfaßt (ohne Angabe) 1 = Großbuchstaben 2 = Kleinbuchstaben
Versch	:	Parameter erlaubt das horizontale Verschieben 0 = mit Verschieben (ohne Angabe) 1 = kein Verschieben erlaubt
PWort	:	Festgelegtes Zeichen für das Passwort

ANMERKUNG Die Systemvariablen ***ewCar***, ***ewTouche***, ***ewScanCode***, sowie die einer Taste zugeordnete Variable ***VarName*** werden bei jedem Anschlag eines Zeichens in einem •`TasteEdit`() auf den neuesten Stand gebracht. Ein •`TasteEdit`() kann ausgewählt werden durch einen Klick oder durch den Befehl •`TasteFocus`(). Eine Überladung des Fensters kann geklärt werden durch den Befehl •`Aktion`(). Der Befehl •`TasteFont`() ermöglicht die Auswahl der Schriftart (Font) der Editier-Fenster-Taste. Die Variable ***ewPosEdit*** beinhaltet die Position des Cursors in der ausgewählten Editier-Taste. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden. Der Befehl •`TasteLösch`() ermöglicht das Löschen einer Editier-Fenster-Taste.

BEISPIEL

```
[Feld:='edi1|edi2|edi3|edi4|']
[C:=0, EditAktiv:='edi1']
•TasteFont(Arial,18)
•Füllfarbe(255) •Pinsel(249)
•TasteEdit(edi1,Texte initial1,100,100,100,20,10,1,1)
•TasteEdit(edi2,Texte initial2,250,100,140,20,10,,2)
•TasteEdit(edi3,Texte initial3,450,100,100,20)
•TasteEdit(edi4,Texte initial4,600,100,110,20)
•TasteFocus(edi1)
•Wenn([EditAktiv<>ewControle],=+3)
[C:=chaind(Feld,ewControle,'|')-1]
[EditAktiv:=ewControle]
•Wenn([ewtouche=9],=-3) ••Tab
[C:=mod(C+1,4),edit:=chamot(Feld,C+1,'|')]
•TasteFocus([edit])
[ewTouche:=0]
•Sprung(=-7)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnStatic) Befehl: TasteFix

Definieren einer statischen Taste

SYNTAX •**TasteFix(*VarName,Typ,Xo,Yo,B,H,Param1,Param2*)**

BESCHREIBUNG Definieren einer statischen Taste

VarName	:	Name einer globalen oder lokalen Variablen, die einer Taste zugeordnet ist
Typ	:	Type der Taste
Xo,Yo	:	Koordinaten der linken oberen Ecke
B	:	Breite der Taste
H	:	Höhe der Taste
Param1,Param2	:	Optionelle Funktions-Parameter

Typ	**Aussehen**
1	Anzeigen des gesetzten Textes in ***Param1***
2	Anzeigen des gesetzten Textes in ***Param1*** in einem 3D Rahmen
3	Anzeigen des aktiven virtuellen Bildschirms
4	Anzeigen des aktiven virtuellen Bildschirms in einem 3D Rahmen
5	Horizontaler Strich
6	Vertikaler Strich
7	Rahmen
8	Rahmen
9	Rahmen
10	Rahmen
11	Rahmen
12	Rahmen
13	Rahmen
14	Rahmen
15	Rahmen
16	Rahmen
17	Rahmen
18	Rahmen
19	Rahmen

ANMERKUNG Die statische Taste kann nur auf der Seite einer Register-Taste verwendet werden. Die 2 optionellen Parameter ***Param1*** und ***Param2*** werden für die Typen 1, 2, 3 und 4 verwendet. Für die Typen 1 und 2 beinhaltet ***Param1*** den Text der geschrieben wird und ***Param2*** wird nicht verwendet. Für die Typen 3 und 4 beinhalten ***Param1*** und ***Param2*** die Koordinaten der linken oberen Ecke der Oberfläche, die aus einen virtuellen Bildschirm gelesen wird. (0,0 ohne Angabe).

BEISPIEL

```
•RegSeite(TasteReg,2)
•TasteFix(O2_1,9,10,20,200,200)
•TasteFix(O2_2,2,20,20,100,20,TEXT)
•VBschLösch(1)•VBschDef(200,200)•Bsch(1)
•BschLösch(236)•Vektor(10,100,100,450)
•TasteFix(O2_3,4,50,50,100,100)
•Bsch(0)
•RegSeite()
```

SIEHE AUCH **•TasteReg(), •RegSeite()**

Alias: (BtnFocus) Befehl: TasteFocus

Auswählen einer Taste (in den Brennpunkt setzen)

SYNTAX •**TasteFocus(*VarName*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Auswahl (in den Brennpunkt setzen) einer Taste von Typ •**Taste**(), •**TasteGrafik**(), •**TasteEdit**()

VarName : Name der globalen oder lokalen Variablen, die mit der Taste verbunden ist.

BEISPIEL

```
•TasteEdit(Edit1,100,100,100,20)
•TasteFocus(Edit1)
```

SIEHE AUCH •**TasteEdit**(), •**TasteGrafik**(), •**Taste**()

Alias: (PolBtn) Befehl: TasteFont

Definieren des Font (Schriftart) für die Windows-Tasten

SYNTAX •**TasteFont(*Schrift,Größe,Stärke,Italic*)**

BESCHREIBUNG Lädt eine Schrift, die für Windows-Tasten verwendet wird.

Schrift	:	Name der Schrift
Größe	:	Höhe der Buchstaben in Pixel
Stärke	:	Stärke der Buchstaben (0 bis 2)
Italic		1 = kursiv (Italic), 0 = normal

Stärke	**Strichstärke:**
0	= Fein
1	= Normal
2	= Fett

ANMERKUNG Die Standardeinstellung für die Stärke ist 1 und die für Italic ist 0.

BEISPIEL

```
•TasteFont(Winddings,40,2)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnProgress) Befehl: TasteFortschr

Definition einer Taste mit einer Darstellung des Fortschritts

SYNTAX •**TasteFortschr(*VarName,Xo,Yo,B,H,Max*)**

BESCHREIBUNG Definition einer Taste mit einer Darstellung des Fortschritts

VarName	:	Name einer globalen oder lokalen Variablen, die einer Taste zugeordnet wird
Xo, Yo	:	Koordinaten der linken oberen Ecke
B	:	Breite der Taste
H	:	Höhe der Taste
Max	:	Anzahl der Positionen die durch den Indikator angezeigt werden

ANMERKUNG Jede neue Position des Indikators für den Fortschritt wird mit Hilfe des Befehls •**TastePos**() übertragen. Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist die Taste lokal an das Modul gebunden, durch das es erzeugt wurde und verschwindet bei der Verkettung mit einem anderem Modul. Es können bis zu 600 Tasten unterschiedlichster Art gleichzeitig verwaltet werden.

Der Befehl •**TasteLösch**() ermöglicht das Löschen einer Taste.

BEISPIEL

```
•TasteFortschr(Prozent,500,200,150,20,100)
[i:=1]
•TastePos(Prozent,[i])[i:=i+1] •Pause(0.2)
•Schleife(99,=-1)
```

SIEHE ALSO •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnImg) Befehl: TasteGrafik

Definieren einer Grafik-Taste

SYNTAX •**TasteGrafik(*VarName,Xo,Yo,Breite,Höhe,Xg,Yg*)**

BESCHREIBUNG Definiert und erstellt eine Grafiktaste, deren obere linke Ecke die Koordinaten ***Xo,Yo*** hat und deren Breite ***Breite*** und Höhe ***Höhe*** ist. Die Koordinaten ***Xg*** und ***Yg*** bezeichnen die linke obere Ecke der Grafik, die die Grafiktaste darstellt.

VarName	:	Name einer lokalen oder globalen numerischen Variablen, die dem Grafikfeld zugeordnet ist
Xo,Yo	:	Koordinaten der oberen linken Ecke
Breite	:	Breite des Knopfes
Höhe	:	Höhe des Knopfes
Xg,Yg	:	Koordinaten der linken oberen Ecke der Grafik, die die Grafiktaste darstellt

ANMERKUNG Sobald die Taste angeklickt wird, wird der Variablen ***VarName*** der Wert 1 zugewiesen. Bei der Rückkehr zum Interventionspunkt des Schülers wird der Wert der Variablen wieder 0.. Die Koordinaten ***Xg,Yg*** zeigen die obere linke Ecke einer Grafik an, die die 4 möglichen Zustände einer Grafiktaste anzeigt. Diese 4 Darstellungsformen müssen horizontal nebeneinander angeordnet sein.

Die vier möglichen Zustände einer Grafiktaste sind:
normal - gedrückt - ausgewählt - inaktiv

Der Befehl •**TasteModus**() ermöglicht es, die Grafiktaste in einen der vier Zustände zu versetzen. Man kann 600 verschiedene Tasten zugleich erzeugen. Der Befehl •**TasteLösch**() ermöglicht das Löschen der Grafik

BEISPIEL

```
••Erzeugen einer Grafiktaste mit den Maßen 64x64
••Erzeugen eines virtuellen Bildschirmes
•VBschDef(256,64)•Bsch(1)
••Laden der Grafik, die die Taste darstellt
•Objekt(Taste,0,0)••Erzeugen der Taste
•TasteGrafik(TasteNeu,320,200,64,64,0,0)
••Freigabe des virtuellen Bildschirms
•VBschLösch(1)••Taste in einen bestimmten Zustand setzen
•TasteModus(TasteNeu,2)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnAide) Befehl: TasteHilfe

Definition eines kleinen temporären Hilfe-Fensters

SYNTAX •**TasteHilfe(*TasteVar,Text,Dauer*)**

BESCHREIBUNG Definition eines kleinen Hilfe-Fensters, welches dann erscheint, wenn die Maus eine bestimmte Zeit (Sekunden) auf der assoziierten Taste verweilt.

TasteVar	:	Name einer mit der Taste assoziierten globalen oder lokalen Variablen
Text	:	Text, der im Hilfefenster erscheint
Dauer	:	Dauer der Darstellung des Hilfefensters (5 Sekunden Standard, maximal 32 Sekunden)

ANMERKUNG Man kann für alle Tasten Hilfefenster erstellen, außer für Menu-Tasten.

BEISPIEL

```
•Taste(Weiter,Weiter,500,10,50,40)
•TasteHilfe(Weiter,'Für weiter, klicken Sie auf Weiter',10)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**() •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteLösch**() •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnInfo) Befehl: TasteInfo

Ermitteln der Parameter einer Taste

SYNTAX •**TasteInfo(*VarName,Xo,Yo,B,H,Fen,Typ,Stil,Zustand,Seg,Pos*)**

BESCHREIBUNG Übernehmen der Erstellparameter einer Taste (Taste, Zone,...)

VarName	:	Name der Variable, die mit einer Taste verbunden wird
Xo, Yo	:	Name der Variablen, die die Koordinaten der linken oberen Ecke der Taste enthalten
Breite	:	Name der Variablen, die die Breite der Taste enthält
Höhe	:	Name der Variablen, die die Höhe der Taste enthält
Fen	:	Name der Variablen, die die Nummer des Fensters der Taste enthält
Typ	:	Name der Variablen, die den Typ der Taste enthält. Diese kann Wert zwischen 0 und 14 enthalten. 0 = Das ist keine Taste 1 = Taste oder TasteRadio oder TastAuswahl 2 = TasteGrafik 3 = Zone 4 = TasteScroll 5 = TasteMenü 6 = TasteListe 7 = Hyp 8 = TasteCombo 9 = TasteReg (nur Version Windows 95) 10 = TasteFortschr (nur Version Windows 95) 11 = TasteCursor (nur Version Windows 95) 12 = TasteStatic (nur Version Windows 95) 13 = TasteEdit 14 = TasteMCI
Stil	:	Name der Variablen, die den Stil der Taste enthält (nur für den Typ 1 und 2). Siehe auch •**TasteModus**()
Zustand	:	Name der Variablen, die den Zustand der MCI-Sequenz enthält, die miut der Taste verbunden ist (nur für den Typ 14 •**TasteMCI**()). Die Werte sind: 0 = Beendet 1 = Pause 2 = Spielt 3 = Schleife
Seg	:	Name der Variablen, die die Nummer des aktiven Segmentes der mit der Taste verbundenen MCI Sequenz enthält
Pos	:	Nummer der Variablen, die die augenblickliche Position in einer MCI Sequenz , die mit der Taste verbunden ist, enthält (nur für den Typ 14 •**TasteMCI**()). Für WAV und MIDI- Sequenzen. ist die Einheit Millisekunden.Für AVI und MPEG ist die Einheit das Bild.

BEISPIEL

```
•TasteScroll(Horiz,1,500,200,150,20)
•TasteInfo(Horiz,X,Y,B,H,Fen,Typ)
•TasteMCI(AVI1,'Seq1.AVI,/Seq2.AVI/',10,10,,,2,,,,,,,'/')
•TasteInfo(AVI1,,,,,,Fen,,Zustand,Seg,Pos)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**() •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteLösch**() •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnListe) Befehl: TasteListe

Definieren eines Listen-Fensters mit Bildlaufleiste

SYNTAX •**TasteListe(*VarName,ZDat{A},ZDat{E},X,Y,B,H,Trennz,WortNr*)**

BESCHREIBUNG Definiert und erstellt ein Fenster mit den Koordinaten der oberen linken Ecke ***Xo,Yo,*** der Breite ***Breite*** und der Höhe ***Höhe*** und einer Bildlaufleiste, in dem die Zeichenketten der Zeichenkettendatei ***ZDat*** erscheint.

VarName	:	Name einer globalen oder lokalen numerischen Variable, die dem Fenster zugeordnet ist
ZDat{A}	:	Erste Zeichenkette der Zeichenkettendatei
ZDat{E}	:	Letzte Zeichenkette der Zeichenkettendatei
Xo,Yo	:	Koordinaten der oberen linken Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Trennz	:	Trennzeichen zur event. Abteilung der ZKetten
WortNr	:	Nummer des Wortes der Zeichenkette der Datei, welche im Fenster dargestellt ist

ANMERKUNG Wenn ***WortNr*** durchgeführt ist, werden vom Fenster die Worte ***WortNr*** der Zeichenketten der Datei ***ZDat*** dargestellt, dessen Trennzeichen ***Trennz*** ist. Die zugeordnete variable Zeichenkette enthält 3 Wörter, die durch einen vertikalen Balken (¦) getrennt sind. Das erste Wort enthält die Nummer des Items, das sich in der ersten Zeile des Fensters befindet, das zweite Wort enthält die Nummer des Items, welches angeklickt wurde und das dritte Wort schließlich enthält das Item selbst. Die Systemvariable ***ewControle*** enthält den Namen der zugeordneten Variablen, sobald ein Item zweimal angeklickt wird. Die zugeordnete Variable kann global oder lokal sein. Wenn die Variable eine lokale ist, so ist die Listentaste lokal an das Modul gebunden, durch das sie erzeugt wurde und verschwindet anläßlich der Verkettung mit einem anderen Modul. Man kann 600 Fenster zugleich erzeugen.

Der Befehl •**TasteLösch**() ermöglicht das Löschen des Fensters.

BEISPIEL

```
•dir(c:\windows\*.ini,dir)
•TasteListe(DirListe,dir(1),,200,200,150,200)
•Eingabe()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**() •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteLösch**() •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (SupCtrl) Befehl: TasteLösch

Löschen von Tasten, Menüs, Auswahltasten und Zonen

SYNTAX •**TasteLösch(*VarName,VarName....*)**

BESCHREIBUNG Löscht eine oder mehrere Tasten (Tasten, Menüs...)

VarName : Name der Taste

ANMERKUNG Wenn der Befehl ohne Parameter angegeben wird, so werden alle Tasten gelöscht (Tasten und Zonen).

Man kann auch eine Liste von Tasten angeben, wobei die Namen durch Beistriche getrennt werden müssen.

BEISPIEL

```
•TasteLösch(Hilfe)
•TasteLösch(Hilfe,Weiter,Zone1,Zone2)
•TasteLösch()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (SupCtrlSauf) Befehl: TasteLöschA

Löschen aller Tasten, Menüs, Auswahltasten und Zonen, außer jene, die aufgelistet sind

SYNTAX •**TasteLöschA(*VarName,VarName....*)**

BESCHREIBUNG Löscht eine oder mehrere Tasten (Tasten, Menüs...), außer jene, deren verknüpfte Variablennamen in der Liste aufscheinen.

VarName : Name der Taste

ANMERKUNG Wenn der Befehl ohne Parameter angegeben wird, werden alle Tasten gelöscht (Tasten und Zonen).

BEISPIEL

```
•TasteLöschA(Hilfe)
•TasteLöschA(Hilfe,Weiter,Zone1,Zone2)
•TasteLöschA()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnMCI) Befehl: TasteMCI

Definieren einer Multimedia-Taste

SYNTAX •`TasteMCI(VarName,Dat,X,Y,B,H,Bef,Rahmen,Anp,Balken,Titel,A,E,Trennz)`

BESCHREIBUNG Definiert und erstellt eine Multimedia-Taste

VarName	:	Name einer globalen oder lokalen Variablenkette, die mit einer MCI-Taste verbunden ist
Datei	:	Name der Datei die abgespielt wird
X,Y	:	Koordinaten der linken oberen Ecke
B	:	Breite des Fensters
H	:	Höhe des Fensters
Bef	:	Bestimmende Parameter für die Art des Sequenz-Beginnes 0 = nach Halt zum Anfang der Sequenz (Standardeinstellung) 1 = Abspielen der Sequenz 2 = Spielen der Sequenz in einer Schleife
Rahmen	:	Stil des Fensterrahmens der MCI-Taste 0 = keinen Rahmen (Standardeinstellung)) 1 = dünner Rahmen 2 = veränderbarer Rahmen
Anp	:	Bestimmende Parameter für die Art der Anpassung der Animation am Fenster 0 = Keine Anpassung der Größe oder der Animation (Standardeinstellung) 1 = Anpassung des Fensters an der Animation 2 = Anpassung der Animation am Fenster
Balken	:	Bestimmende Parameter für das Aussehen des Kontrollbalkens der MCI-Taste 0 = Kein Kontrollbalken (Standardeinstellung) 1 = Anzeigen des Balkens ohne Tastenmenü 2 = Anzeigen des Balkens mit Tastenmenü (ohne die Option Datei/Öffnen) 3 = Anzeigen des Balkens mit Tastenmenü mit allen Optionen
Titel	:	Bestimmende Parameter für das Aussehen des Titels im Fenster der MCI-Taste 0 = Kein Titel (Standardeinstellung) 1 = Zeigt im Titel den Dateinamen an 2 = Zeigt im Titel die Position in der Sequenz an 3 = Zeigt im Titel den Status der Sequenz an

ANMERKUNG Die zugeordnete Zeichenkettenvariable beinhaltet dafür eine Nummer, welche einen automatische Alias-Namen beinhaltet, der die Steuerung der Sequenz mit Hilfe des Befehls •**MCI**() ermöglicht. Es ist für die Tonsequenzen nicht notwendig die Parameter ***X, Y, B, H*** zu setzen, wenn man sie am Kontrollbalken nicht anzeigen möchte. Im Modus 256 Farben ist die Anpassung an die laufende Palette der Sequenz automatisch, die Verwendung des Befehls •**FarbeFrei**() ist nicht mehr notwendig. Der Befehl •**TasteLösch**() ermöglicht das Löschen einer MCI-Taste.

BEISPIEL

```
•TasteMCI(AliasWav,ding.wav,,,,,2)
•MCI(volume [AliasWav] 50000 50000)
•TasteMCI(AliasMid,passport.mid,10,10,,,0,,,1)
•MCI(volume [AliasWav] 50000 50000)
•TasteMCI(AliasAvi1,seqavi1.avi,400,10,,,2)
•TasteMCI(AliasAvi2,seqavi2.avi,400,10,,,2,2,,2,1)
•MCI(SET [AliasAvi1] SPEED 500)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnMenu) Befehl: TasteMenü

Definieren einer Menüleiste

SYNTAX •**TasteMenü(*VarName,´Item1,...ItemN´*)**

BESCHREIBUNG Definiert und erstellt am oberen Rand des Bildschirmes eine Menüleiste

VarName : Name einer globalen oder lokalen Variable vom Typ Zeichenkette, die dem Menü zugeordnet ist

´Item1,...ItemN´ : Items der Menüleiste zwischen Apostrophen

ANMERKUNG Die zugeordnete Zeichenkette enthält zwei Worte, die durch den vertikalen Balken (¦) getrennt werden. Das erste Wort enthält die Nummer des angeklickten Items und das zweite das angeklickte Item selbst.

Die Liste der Items muß zwischen Apostrophen geschrieben werden. Jedes Item kann ein Untermenü generieren, indem die Items des Untermenüs zwischen geschwungene Klammern geschrieben werden. Wenn vor einem Buchstaben das Zeichen "&" geschrieben wird, so wird dieser Buchstabe unterstrichen.

Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist die Menüleiste lokal an das Modul gebunden, durch das sie erzeugt wurde und verschwindet anläßlich der Verkettung mit einem anderen Modul.

Man kann nur eine Menüleiste erzeugen.Der Befehl •**TasteLösch**() ermöglicht das Löschen des Fensters.

BEISPIEL

```
•TasteMenü(Menü,´&Datei{&Neu,&Öffnen,E&xit},&Optionen´)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnMenuStyle) Befehl: TasteMenüStil

Definieren eines Item-Stils in einem Menü

SYNTAX •**TasteMenüStil(*VarMenü,NumItem,Stil*)**

BESCHREIBUNG Legt den Stil eines Items (unterlegt, aktiv und/oder ausgewählt und/oder nicht ausgewählt) in einem Menü fest.

VarMenü	:	Name der mit dem Menü verbundenen Variablen
NumItem	:	Nummer des Items, dessen Stil man ändern möchte
Stil	:	Stil, welchen das Item annehmen kann 0 = unterlegt 1 = aktiv 2 = ausgewählt 3 = nicht ausgewählt

BEISPIEL

```
•TasteMenü(Menü,'&Datei{&Neu,&Öffnen,&Speichern,Beenden}')
•TasteMenüStil(Menü,3,0)
•TasteMenüStil(Menü,3,3)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnStyle) Befehl: TasteModus

Definieren des Tasten-Modus

SYNTAX •**TasteModus(*VarName,Modus,HRicht,VRicht,Opt*)**

BESCHREIBUNG Definiert den Zustand einer Taste. Es wird die Art festgelegt, in der die Taste am Bildschirm erscheint.

VarName : Name einer numerischen Variablen, die dem Feld zugeordnet ist

Modus : Modus der Taste (Werte zwischen 1 und 3)

Modus **Zustand der Taste**
1 = normal
2 = ausgewählt
3 = inaktiv

Hricht : Horizontale Position des Textes in der Taste
0 = Zentriert (Standardeinstellung)
1 = Ausrichtung links
2 = Ausrichtung rechts

Vricht : Vertikale Position des Textes in der Taste
0 = Zentriert (Standardeinstellung)
1 = Ausrichtungt oben
2 = Ausrichtung unten

Opt : Option für das Aussehen der Taste
0 = Text in einer Zeile (Standardeinstellung)
1 = Text auf meheren Zeilen
2 = Gibt einer Radio- und AuswahlTaste das Aussehen einer Drucktaste

ANMERKUNG Der Modus ausgewählt ist nur für Grafiktasten, Radio- und Auswahltasten gültig!

Die Parameter ***HRicht***, ***VRicht*** und ***Opt*** werden sowohl vom 16bit, als auch vom 32bit Interpreter unterstützt, aber unter Windows 95. Wenn ***VarName*** fehlt, dann beziehen sich die Parameter ***HRicht***, ***VRicht*** und ***Opt*** auf alle Tasten, die nach diesem Befehl erstellt werden. Wenn ein Variablen-Name ***VarName*** gesetzt ist, dann beziehen sich die drei Parameter nur auf die Taste ***VarName***.

BEISPIEL

```
••Erzeugen einer Grafiktaste mit den Maßen 64x64
••Erzeugen eines virtuellen Bildschirms
•VBschDef(256,64) •Bsch(1)
••Laden der Grafik, die Taste repräsentiert
•Objekt(Taste,0,0)
••Erzeugen einer Grafiktaste
•TasteGrafik(Bgraph,320,200,64,64,0,0)
••Freigabe des virtuellen Bildschirms
•VBschLösch(1)
•TasteModus(Bgraph,2) ••Ausgewählter Zustand
•Taste(B1,Text auf mehreren Zeilen,10,10,100,100)
•TasteModus(B1,,,,1)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (NomBtn) Befehl: TasteName

Ermitteln des Namens einer aktiven Taste (Menü, Auswahltaste oder Zone)

SYNTAX •**TasteName(*Nummer,TastenName*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln des Namens einer Variablen, einer Zone oder einer aktiven Taste.

Nummer : Nummer der Zone oder der Taste, von der man den Namen der verbundenen Variablen wissen möchte

TastenName : Name der Zeichenkette, welche den Namen der verbundenen Variablen der Taste oder Zone enthält

BEISPIEL

```
•• ermittelt die Anzahl der aktiven Tasten
•TasteZahl(AnzTasten)
•Wenn([AnzTasten=0])•Ende()
[i:=1]
•• ermittelt und stellt den Namen der Taste dar
•TasteName([i],Name) [Name]
[i:=i+1]
•Schleife([AnzTasten-1],=-2)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnPos) Befehl: TastePos

Positionieren des Cursors bei TasteListe, TasteScroll, TasteCombo und TasteComboEdit

SYNTAX •**TastePos(*VarName,Pos*)**

BESCHREIBUNG Dieser Befehl erlaubt das Positionieren des Cursors bei Tasten vom Typ •**TasteCombo**(), •**TasteComboEdit**(), •**TasteListe**() oder •**TasteScroll**() an der gewünschte Position Pos.

VarName	:	Name der globalen oder lokalen Variablen, welche mit TasteListe, TasteScroll, TasteCombo oder TasteComboEdit verbunden ist
Pos	:	Position des Cursors

BEISPIEL

```
•TasteScroll(HScroll,1,100,100,200,20,100)
•TastePos(HScroll,25)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnRadio) Befehl: TasteRadio

Definieren einer Windows Radio-Taste

SYNTAX •**TasteRadio(*VarName,Text,Xo,Yo,Breite,Höhe,+*)**

BESCHREIBUNG Definiert und erstellt eine Radiotaste mit den Koordinaten der linken oberen Ecke ***Xo,Yo***, der Breite ***Breite*** und der Höhe ***Höhe***.

VarName	:	Name einer globalen oder lokalen numerischen Variablen, die dem Feld zugeordnet ist
Text	:	Text, der im Feld erscheint
Xo,Yo	:	Koordinaten der linken oberen Ecke
Breite	:	Breite des Feldes
Höhe	:	Höhe des Feldes
+	:	Wenn + angegeben wird, so kennzeichnet dies den Beginn einer Gruppe von Radiotasten

ANMERKUNG Sobald das Feld angeklickt wird, wird der zugeordneten Variablen ***VarName*** der Wert 1 zugewiesen, sofern das Feld gültig ist, sonst den Wert 0.

Die zugeordnete Variable kann global oder lokal sein. Handelt es sich um eine lokale Variable, so ist die Radiotaste lokal an das Modul gebunden, durch das es erzeugt wurde und verschwindet anläßlich der Verkettung mit einem anderen Modul. Wenn im Text ein Beistrich oder ein Apostroph vorkommt, so sind sie zwischen Apostrophen zu schreiben. Man kann insgesamt 600 Tasten erzeugen.

Der Befehl •**TasteLösch**() ermöglicht das Löschen des Fensters.

BEISPIEL

```
•Pinsel(236)•Füllfarbe(255)
•TasteRadio(Opt1,Option 1,500,10,50,40,+)
•TasteRadio(Opt2,Option 2,500,10,50,40)
•TasteRadio(Opt3,Option 3,500,10,50,40)
•TasteModus(Opt1,2)    ••Auswahl der ersten Taste
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TastePos**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnOnglet) Befehl: TasteReg

Definition einer Register-Taste

SYNTAX •**TasteReg(*VarName,'Reg1,...,RegN',Xo,Yo,Breite,Höhe,Modus*)**

BESCHREIBUNG Definiert und erstellt eine Register-Taste

VarName	:	Name einer globalen oder lokalen Variablenkette, die einer Taste zugeordnet wird
'Reg1, ,RegN'	:	Texte der Winkel-Tasten
Xo, Yo	:	Koordinaten der linken oberen Ecke
Breite	:	Breite der Taste
Höhe	:	Höhe der Taste
Stil	:	Stil der Register-Taste (Zwischen 1 und 8) 1 = Register von variabler Größe (Standardeinstellung) 2 = Register von fixer Größe 3 = Register erscheinend auf mehreren Linien 4 = 2 und 3 5 = Register in Form einer Taste 6 = Register in Form einer Taste von fixer Größe 7 = Register in Form einer Taste von fixer Größe auf mehreren Linien erscheinend 8 = 6 und 7

ANMERKUNG Dieser Befehl erstellt nur die Hülle von einer Register-Taste. Um die den Registern zugeordneten Seiten zu füllen, muß man den Befehl •**RegSeite**() verwenden. Dieser ermöglicht die Auswahl einer Seite der Register-Taste und das Zuweisen von allen Arten von Tasten (außer dem Menü)

BEISPIEL

```
•TasteReg(TR,'Register1,Register2,Register3,10,10,400,470)
•RegSeite(TR,1)
•TasteFix(R1_1,13,98,38,204,204)
•TasteLösch(1)•VBschDef(400,400)•Bsch(1)
•Bild(Portrait.pcx,,0,0,200,200,,3)
•TasteFix(R1_2,3,100,40,200,200)
•Bsch(0)•Pinsel(236)•Füllfarbe(245)
•TasteFix(R1_3,7,10,250,380,210)
•TasteFix(R1_4,6,195,255,3,200)
•TasteFix(R1_5,2,15,240,83,20,'Titel links)
•TasteFix(R1_6,1,307,240,75,20,'Titel rechts)
•TasteListe(R1_7,'L1,L2,L3,L4,L5',30,270,150,78,',')
•TasteListe(R1_8,'L1,L2,L3,L4,L5',215,270,150,78,',')
•Taste(R1_10,'Taste Links',80,354,100,40)
•Taste(R1_11,'Taste Rechts',265,354,100,40)
•TasteCursor(R1_12,1,30,410,150,40,20,0,1)
•TasteCursor(R1_13,1,215,410,150,40,20,0,0)
••========== Register Seite 2 ==============
•RegSeite(TR,2)
•TasteFix(R2_1,2,10,40,83,20,'ANMERKUNG')
•TasteFix(R2_2,7,10,60,380,210)
•Füllfarbe(246)•TasteFont(andante,14)
•TasteFix(R2_3',2,20,70,360,190, Text)
•TasteFix(R2_4,7,10,280,380,120)
•Taste(R2_5,ABBRECHEN,210,420,100,40)
••========== Register Seite 3 ==============
•RegSeite(TR,3)
•TasteAuswahl(R3_1,Die hier!,80,60,100,20)
•TasteAuswahl(R3_2,Die dort!,80,90,100,20)
•TasteAuswahl(R3_3,Nein dort!,80,120,100,20)
•TasteAuswahl(R3_4,Ja hier!,80,150,100,20)
•TasteAuswahl(R3_5,Oder hier!,80,180,100,20)
•TasteAuswahl(R3_6,Oder dort unten!,80,210,100,20)
•RegSeite()
```

SIEHE AUCH

•**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TastePos**(), •**TasteScroll**(), •**Zone**()

Alias: (BtnDefile) Befehl: TasteScroll

Definieren einer Bildlaufleiste

SYNTAX •**TasteScroll(*VarName,Typ,X,Y,Breite,Höhe,Stufen*)**

BESCHREIBUNG Definiert und erstellt eine Bildlaufleiste (Scrolltaste) mit den Koordinaten ***X,Y*** der oberen linken Ecke, mit der Breite ***Breite*** und der Höhe ***Höhe***. ***Typ*** definiert die Ausrichtung der Bildlaufleiste.

VarName	:	Name einer globalen oder lokalen Variablen, die der Bildlaufleiste zugeordnet ist
Typ	:	Ausrichtung der Bildlaufleiste; 1 = horizontale Leiste 2 = vertikale Leiste
X,Y	:	Koordinaten der linken oberen Ecke
Breite	:	Breite der Leiste
Höhe	:	Höhe der Leiste
Max	:	Maximaler Wert der extremsten Position des Cursors (100 als Standardeinstellung und 32536 maximal)
Min	:	Minimaler Wert der Initialisierungsposition des Cursors (0 als Standardeinstellung und 32536 maximal)

ANMERKUNG Jeder Klick auf einen Punkt der Bildlaufleiste erhöht oder verringert die zugeordnete Variable ***VarName***. Die zugeordnete Variable kann eine globale oder eine lokale sein. Wenn es sich um eine lokale Variable handelt, so ist auch die Bildlaufleiste lokal an das Modul, das es erzeugt, gebunden und verschwindet bei der Verkettung mit einem anderen Modul. Umgekehrt erscheinen die Werte Min und Max des Cursors von seiner Initialisierungsposition. Man kann gleichzeitig bis zu 600 Tasten unterschiedlichster Art erzeugen.

Der Befehl •**TasteLösch**() ermöglicht das Löschen der Bildlaufleiste.

BEISPIEL

```
•TasteScroll(Horiz,1,500,200,150,20)
•TasteScroll(Verti,2,700,10,20,180,50)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TastePos**(), •**TasteRadio**(), •**Zone**()

Alias: (NbBtn) Befehl: TasteZahl

Ermitteln der Anzahl der verwendeten Tasten, Menüs, Auswahltasten und Zonen

SYNTAX •***TasteZahl(Anzahl)***

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln der Anzahl aller aktiven Tasten (Tasten, Menüs...).

Anzahl : Name der numerischen Variablen, welche die Anzahl der aktiven Tasten, Zonen ... übernimmt.

BEISPIEL

```
•TasteZahl()
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**(), •**TasteCombo**(), •**TasteComboEdit**(), •**TasteFont**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**(), •**TasteLöschA**(), •**TasteMenü**(), •**TasteMenüStil**(), •**TasteRadio**(), •**TasteScroll**(), •**Zone**()

Alias: (PCurT) Befehl: TCursPos

Ermitteln der Position des Textcursors

SYNTAX •**TCursPos(*X,Y*)**

BESCHREIBUNG Ermittelt die Position des Textcursors und speichert sie in den Variablen ***X*** und ***Y***

X,Y : Koordinaten des Textcursors

BEISPIEL

```
•Font(Arial,16)
•Pos(100,100) UMRAHMTER TEXT
•TCursPos(CurX,CurY)
•Rechteck(100,100,[CurX],[CurY-16])
```

SIEHE AUCH •**GCursPos**()

Alias: (Texte) Befehl: Text

Erstellen von Grafiktext

SYNTAX

```
•Text(X,Y,H,B,Winkel,Text,Schrift,Textfarbe,Hgrundfarbe,
      Stärke,Italic,Durchgestr,Unterstr)
```

BESCHREIBUNG Anzeige des Textes an den Koordinaten ***X,Y*** mit der Höhe ***H***, der Breite ***B***, ausgerichtet nach dem Winkel ***Winkel***, mit der Farbe ***Textfarbe*** auf dem Hintergrund ***Hgrundfarbe*** in der Schrift ***Schrift*** mit der Stärke ***Stärke***, dem Modus ***Italic*** und mit durchgestrichenem und/oder unterstrichenem Text.

X,Y	:	Koordinaten der Textanzeige
H,B	:	Höhe und Breite der Buchstaben
Winkel	:	Anzeigewinkel des Textes (in 10-tel Grad)
Schrift	:	ausgewählte Schrift
Textfarbe	:	Farbe des Textes
Hgrundfarbe	:	Farbe des Hintergrundes
Stärke	:	Stärke der Buchstaben
Italic	:	Italic (1 = Italic, 0 = Roman)
Durchgestr	:	Durchgestrichener Buchstabe
		(1 = durchgestrichener Text, 0 = Normaltext)
Unterstr	:	Unterstrichener Buchstabe
		(1 = unterstrichener Text, 0 = Normaltext)

ANMERKUNG Der Anzeigewinkel des Textes wird in 10-tel Graden ausgedrückt. Um z.B. einen Winkel von 45 Grad zu erhalten, muß man 450 angeben. Nur die "True Type" - Schriften können beliebig ausgerichtet werden. Um eine transparente Schrifthintergrundfarbe zu haben, muß man den Wert **-1** als Wert für die Hintergrundfarbe ***Hgrundfarbe*** angeben.

BEISPIEL

```
•Text(50,400,40,25,450, '45 Grad Text',Arial,255)
```

Alias: (TxtMod) Befehl: TextMod

Aktivieren/Deaktivieren des freien Textmodus

SYNTAX **•TextMod(+/-)**

BESCHREIBUNG Der aktivierte freie Textmodus ermöglicht das Anzeigen des Textes und von Hypertext in einem Kommentarfenster in Kombination mit den Optionen, Schriften, Farbe und Ausrichtungen

•TxtMode(+) : Aktiviert den freien Textmodus

•TxtMode(-) : Deaktiviert den freien Textmodus (Standardeinstellung)

ANMERKUNG Wenn der freie Textmodus deaktiviert ist, funktioniert die Textanzeige wie bei den Versionen vor der Version 21.

BEISPIEL

```
•TextMod(+)
•HypLösch()•HypStil(236,255,6)
•KomFen(100,10,350,500,255,248)
•Font(ARIAL,38,2,1,1)
•Ausrichten(2)Text zentriert, italic, weiß unterstrichen
•Police(ARIAL,25,2,0)
•Pinsel(255,255,0) Text gelb sehr Klein
•Pinsel(0,255,255) •Font(Courier new,38,2,0) und größer
•Font(Arial,28,2,0) Das ist ein kilometerlanger Text, in
mehreren Zeilen ausgerichtet, (•Pinsel(255,255,0)Wörter in
Gelb •Pinsel(0,255,255)) und
der Text beginnt nach einem Wechsel der Farbe.
Das ist eine •Hyp(Reihe von Wörtern in Hypertext)
welche sich über mehrere Zeilen positionieren.
Die Verwaltung ist derzeit •Hyp(automatisch).
•TextMod(-)
```

Alias: (Beep) Befehl: Ton

Ausgeben eines Signal-Tones

SYNTAX •**Ton()**

BESCHREIBUNG Gibt über den Lautsprecher des Computers einen Signal-Ton aus

ANMERKUNG Die Angabe eines Parameters mit einem Wert zwischen 0 und 4 bewirkt die Ausgabe verschiedener von Windows vordefinierter Signaltöne (sofern der Computer mit einer Soundkarte ausgestattet ist).

BEISPIEL

```
•Ton()
•Ton(3)
```

SIEHE AUCH •**MCI**()

Alias: (Transp) Befehl: Transp

Erzeugen einer transparenten Maske

SYNTAX •**Transp(*Bildname,Farbe*)**

BESCHREIBUNG Erzeugt eine Maske des Bildes ***Bildname*** mit der Farbe ***Farbe*** als Transparenzfarbe.

Bildname : Name des Bildes

Farbe : Transparente Farbe

ANMERKUNG Die Erzeugung einer Maske ist notwendig, wenn man eine Grafik auf einen Hintergrund mit transparenter Farbe plazieren will.

BEISPIEL

```
•Transp(Bild.PCX,0)    ••schwarz erscheint transparent
•Objekt(Bild.PCX,100,200,,6)
```

SIEHE AUCH •**Bild**(), •**Objekt**(), •**GrafikFormat**()

Alias: (HMS) Befehl: Uhrzeit

Ermitteln der Systemzeit (Uhrzeit)

SYNTAX •**Uhrzeit(*StundVar,MinVar,SekVar,HstelVar*)**

BESCHREIBUNG Speichert die Systemzeit in den numerischen Variablen ***Stundvar***, ***MinVar***, ***SekVar*** und ***HstelVar*** ab.

StundVar	:	Name der Variablen, in der die Stunde gespeichert wird
MinVar	:	Name der Variablen, in der die Minuten gespeichert werden
SekVar	:	Name der Variablen, in der die Sekunden gespeichert werden
HstelVar	:	Name der Variablen, in der die Hundertstel gespeichert werden

ANMERKUNG Wenn man an stelle der Variablen numerische Werte angibt, so wird die Systemzeit anhand dieser Werte aktualisiert.

BEISPIEL

```
•Uhrzeit(Stunde,Minute,Sekunde,Hundertstel)
Es ist [Stunde]:[Minute]:[Sekunde]:[Hundertstel].
```

SIEHE AUCH •**Datum**()

Alias: (HMSSyst) Befehl: UhrzeitSyst

Stellt die System-Uhrzeit wieder her.

SYNTAX •**UhrzeitSyst()**

BESCHREIBUNG Dieser Befehl erlaubt das wiederherstellen Der System-Uhrzeit, das kann notwendig sein, wenn die Uhrzeit mit Hilfe des Befehls •**Uhrzeit()** geändert wurde

BEISPIEL

```
•UhrzeitSyst()
```

SIEHE AUCH •**Uhrzeit()**

Alias: (InsTrad) Befehl: ÜbersEinf

Einfügen einer Übersetzung in die Anhäufung der Übersetzungen

SYNTAX •**ÜbersEinf(*Übersetzung*)**

BESCHREIBUNG Fügt eine Übersetzung in die Anhäufung der Übersetzungen ein.

Übersetzung : Name der Übersetzung

ANMERKUNG Man kann auch eine Liste an Übersetzungen angeben, wobei die Namen durch einen Beistrich getrennt sein müssen.

Jede Übersetzung wird am Beginn der Anhäufung der Übersetzungen eingefügt.

BEISPIEL

```
•ÜbersEinf(&AIDE)
•ÜbersEinf(&AIDE,&POL,&QQC,&M)
```

SIEHE AUCH •**ÜbersLösch**()

Alias: (SupTrad) Befehl: ÜbersLösch

Löschen von Übersetzungen in der Anhäufung der Übersetzungen

SYNTAX •**ÜbersLösch(*Name,Name,...*)**

BESCHREIBUNG Löscht eine Übersetzung in der Anhäufung der Übersetzungen.

Name : Name einer Übersetzung

ANMERKUNG Wenn der Befehl ohne Parameter angegeben wird, so werden alle Übersetzungen gelöscht. Man kann auch eine Liste der zu löschenden Übersetzungen angeben, wobei die Namen durch einen Beistrich getrennt werden müssen.

BEISPIEL

```
•ÜbersLösch(&HILFE)
•ÜbersLösch(&HILFE,&POL,&QQC,&M)
•ÜbersLösch()          ••Löscht alle Übersetzungen
```

SIEHE AUCH •**ÜbersEinf**()

Alias: (VarCha) Befehl: VarCha

Speichern von allen globalen Variablen in eine Zeichenkettendatei

SYNTAX •**VarCha(*Datei*)**

BESCHREIBUNG Speichert alle globalen Variablen (außer Tabellen) und ihren Inhalt in eine Zeichenkettendatei.

Datei : Name der Zeichenkettendatei

ANMERKUNG Um diese Variablen wiederherzustellen, ist lediglich die Zeichenketten der Datei anzuzeigen, in der sie gespeichert wurden.

BEISPIEL

```
•VarCha(C:\temp\VarA)
•VarLösch()                    ••löscht alle globalen Variablen
[_num:=fnbcha(c:\temp\VarA), _Zeile:=1]
[c:\temp\VarA{_Zeile},_Lin:=_Zeile+1]•Schleife([_num-1,=)
```

SIEHE AUCH •**TabCha**(), •**VarLösch**()

Alias: (SupVar) Befehl: VarLösch

Löschen von Variablen

SYNTAX •**VarLösch(*VarName,VarName,...*)**

BESCHREIBUNG Löscht eine numerische Variable oder Tabelle, die zuvor definiert wurde.

VarName : Name der Variablen

ANMERKUNG Man kann auch eine Liste von zu löschenden Variablen angeben, wobei die Namen durch einen Beistrich getrennt werden müssen.

BEISPIEL

```
•VarLösch(Zähler)
•VarLösch(Zähler,Nummer,Index,Zeiger)
```

SIEHE AUCH •**VarCha**(), •**TabCha**()

Alias: (NomVar) Befehl: VarName

Ermitteln des Namens einer verwendeten Variablen

SYNTAX •**VarName(*Art,Nummer,VarName,Typ*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln des Namens einer verwendeten lokalen oder globalen Variablen.

Art	:	Parameter, der die Art der zu ermittelnden Variablen festlegt (1 = globale Variable, 2 = lokale Variable)
Nummer	:	Nummer der Variablen, deren Namen man ermitteln möchte
VarName	:	Name der Zeichenkettenvariablen, die den Namen der Variablen übernimmt
Typ	:	Name der numerischen Variablen, der den Typ der Variablen enthält. (0 = Numerisch, 1 = Tabelle, 2 = Zeichenkette)

BEISPIEL

```
•VarZahl(VarAnzahl)••ermittelt die Anzahl der Variablen
•Wenn([VarAnzahl=0])•Ende()
[i:=1]
•VarName(1,[i],Name)[Name]••zeigt den Namen der Variablen
[i:=i+1]
•Sprung([VarAnzahl-1],=-2)
```

SIEHE AUCH •**VarZahl**()

Alias: (NbVar) Befehl: VarZahl

Ermitteln der Anzahl der verwendeten Variablen

SYNTAX

- **VarZahl(*Art,Anzahl*)**

BESCHREIBUNG Dieser Befehl ermöglicht das Ermitteln der Anzahl der verwendeten lokalen oder globalen Variablen.

Art	:	Parameter, der die Art der zu ermittelnden Variablen festlegt (1 = globale Variable, 2 = lokale Variable)
Anzahl	:	Name der numerischen Variablen, welche die Anzahl der verwendeten Variablen enthält

BEISPIEL

```
•VarZahl(VarAnzahl)
```

SIEHE AUCH

- **VarName**()

Alias: (DimVirt) Befehl: VBschDef

Definieren eines virtuellen Bildschirmes

SYNTAX •**VBschDef(*Breite,Höhe*)**

BESCHREIBUNG Definiert die Auflösung eines virtuellen Bildschirms

ANMERKUNG Wird nichts anderes angegeben, so haben die virtuellen Bildschirme die Auflösung der aktuellen Module.

Man kann jeden virtuellen Bildschirm mit verschiedenen Auflösungen definieren, indem man der ersten Verwendung des Befehls •**Bsch**() den Befehl •**VBschDef**() voranstellt. Wenn man einen bereits existierenden virtuellen Bildschirm neu dimensionieren will, so muß man ihn zuvor durch den Befehl •**VBschLösch**() unterdrücken.

BEISPIEL

```
•VBschDef(1024,300) •Bsch(1)
•VBschDef(200,200)  •Bsch(2)
```

SIEHE AUCH •**Bsch**(), •**BschKopie**(), •**VBschLösch**()

Alias: (SupEcran) Befehl: VBschLösch

Löschen eines virtuellen Bildschirmes

SYNTAX •**VBschLösch(*BschNummer*)**

BESCHREIBUNG Löscht einen virtuellen Bildschirm.

Bschummer : Nummer des virtuellen Bildschirms, der gelöscht werden soll.

ANMERKUNG Wenn der Befehl ohne Parameter ausgeführt wird, dann werden alle virtuellen Bildschirme gelöscht. Man kann auch eine Liste von zu löschenden Bildschirmen angeben, jede Bildschirmnummer muß durch einen Beistrich getrennt werden.

BEISPIEL

```
•VBschLösch(2)
•VBschLösch(2,3,25,33)
```

SIEHE AUCH •**VBschDef**(), •**Bsch**()

Alias: (Vect) Befehl: Vektor

Zeichnen eines Vektors

SYNTAX •**Vektor(*Xo,Yo,Länge,Alpha,WinkelSpitze,GrößeSpitze*)**

BESCHREIBUNG Zeichnet einen Vektor mit dem Ursprung ***Xo,Yo***, der Länge ***Länge*** mit der Ausrichtung nach dem Winkel ***Alpha***. Der Winkel wird in Grad im trigonometrischen Sinne ausgedrückt.

Xo,Yo	:	Koordinaten des Ursprungs
Länge	:	Länge des Vektors
Alpha	:	Ausrichtung des Vektors (in Grad)
WinkelSpitze	:	Winkel der Öffnung des Pfeilspitze (Standardwert = 90 Grad)
GrößeSpitze	:	Länge der Seiten der Pfeilspitze (Standardwert = 15 Pixel) '

ANMERKUNG Die Darstellung erfolgt mit der Farbe, dem Stil und dem Modus des aktuellen Pinsels.

Das Innere der Pfeilspitze wird mit der Farbe der Füllfarbe ausgefüllt. ***WinkelSpitze*** und ***GrößeSpitze*** haben einen Standardwert von 90 Grad und 15 Pixel.

BEISPIEL

```
•Vektor(320,200,150,45)
```

SIEHE AUCH •**Nadel**(), •**Füllfarbe**(), •**Pinsel**(), •**ZeiMod**(), •**PinselStil**()

Alias: (Suite) Befehl: Weiter

Verbinden mit dem nächsten Modul

SYNTAX **•Weiter(*Modulname,+/-,p1,p2,...,pn*)**

BESCHREIBUNG Verläßt das aktuelle Modul und führt das nächste Modul ***Modulname*** aus.

Modulname	:	Name des Moduls
+/-	:	Option für das Speichern der Anhäufung der Übersetzungen
p1,...pn	:	Parameter beim Übergang zum Modul

ANMERKUNG Standardmäßig wird die Anhäufung der Übersetzungen beibehalten. Um die Anhäufung der Übersetzungen zu löschen, ist das Zeichen ´-´ als zweiter Parameter anzugeben.

Anläßlich einer Verbindung mit dem nächsten Modul werden alle lokalen Aufgaben beendet. Das aufgerufene Modul kann die übernommenen Parameter wiedererlangen und zwar mit dem Befehl **•ModParam**(). Es können bis zu 32 Parameter übernommen werden.

BEISPIEL

```
•Weiter(Modul2,-)••Verbindung mit dem Modul2 und Löschen
                  ••der angehäuften Übersetzungen
•Weiter(Modul3,,[x+2],20,Demonstration,'•Pinsel(255)',10)
```

SIEHE AUCH **•ModParam**(), **•CoMod**(), **•ModVerb**(), **•Ende**(), **•Aufgabe**()

Alias: (Si) Befehl: Wenn

Bedingungen im Text- oder im Kommentarteil

SYNTAX •**Wenn(*[Ausdruck],Zeile*)** oder •**Wenn(*[Ausdruck],Label*)**

BESCHREIBUNG Dieser Befehl ermöglicht die Ausführung einer Text- oder Kommentarzeile von Bedingungen abhängig zu machen. Wenn die Bedingung erfüllt wird, so wird die Zeile ausgeführt; wenn nicht, so wird zur nächsten Zeile gesprungen.

Die Bedingung muss immer zwischen eckigen Klammern stehen. Optional kann auch ein zweiter Parameter - eine Zeilennummer oder ein Label- angegeben werden, der die auszuführende (zu überspringende Zeile) angibt, wenn die Bedingung zutrifft (nicht zutrifft). Diese Option erlaubt eine Ausweitung des Aktionsfeldes von •**Wenn**() auf mehrere Zeilen.

Ausdruck : Bedingung

Zeile oder

Label : Nummer oder Label der Zeile

ANMERKUNG Der Befehl •**Wenn()** kann nur im Textteil oder Kommentarteil vorkommen.

Das Äquivalent für den Antwortteil ist der Befehl •**BedAM**(). Wenn die Zeilennummer oder der Label durch das Zeichen = (Gleichheitszeichen) ersetzt wird, so gilt die Bedingung für die aktuelle Zeile. Dem Zeichen = kann eine Anzahl negativer oder positiver Zeilen folgen, relativ zur Zeile im Kurs. Zum Beispiel •**Wenn([Ausdruck],=+5)**, dann springt man 5 Zeilen, gezählt ab der Zeile, in der die Bedingung nicht erfüllt wurde, weiter.

BEISPIEL

```
•Wenn([Zähler < 3]) Fahren Sie fort!
•Wenn([Motor=2],Falsch) [Motor=3]
•Objekt(Band,0,460)
[Folge:=´Mod3´]
Falsch    [Motor:=Motor+1]
```

SIEHE AUCH •**BedAM**() und die Funktionen des Rechners

Alias: (HRideau) Befehl: WipeHor

Anzeigen von Bildern in Wippenform (Vorhang)

SYNTAX •**WipeHor(*BschV,Xsv,Ysv,Xdh,Ydh,Xdb,Ydb,Etappen,Modus*)**

BESCHREIBUNG Anzeige eines Teiles oder eines ganzen virtuellen Bildschirms in Form eines Vorhangs (Wipe).

BschV	:	Nummer des virtuellen Bildschirms
Xsv,Ysv	:	Koordinaten der oberen linken Ecke der Oberfläche des virtuellen Bildschirms, die kopiert wird
Xdh,Ydh		
Xdb,Ydb	:	Koordinaten der Diagonale der Oberfläche des physischen Bildschirms, auf den die Oberfläche des virtuellen Bildschirms kopiert wird
Etappen	:	Anzahl der Etappen, um das Bild am Bildschirm anzuzeigen
Modus		**Modus der Anzeige** **0** = Verschwindenlassen des sichtbaren Bildes nach links durch Aufdecken des Inhalts des virtuellen Bildschirms **1** = Verschwindenlassen des sichtbaren Bildes nach rechts durch Aufdecken des Inhalts des virtuellen Bildschirms **2** = Verschwindenlassen des sichtbaren Bildes nach unten durch Aufdecken des Inhalts des virtuellen Bildschirms **3** =Verschwindenlassen des sichtbaren Bildes nach oben durch Aufdecken des Inhalts des virtuellen Bildschirms **5** = Verschwindenlassen des sichtbaren Bildes nach links und rechts durch Aufdecken des Inhalts des virtuellen Bildschirms **6** = Erscheinenlassen des Inhalts des virtuellen Bildschirms und verschwindenlassen des sichtbaren Bildes nach links und rechts **7** = Verschwindenlassen des sichtbaren Bildes nach oben und unten durch Aufdecken des Inhalts des virtuellen Bildschirms **8** = Erscheinenlassen des Inhalts des virtuellen Bildschirms und verschwindenlassen des sichtbaren Bildes nach unten und oben **10** = Erscheinenlassen des Inhalts des virtuellen Bildschirms durch verschieben des sichtbaren Bildes nach rechts **11** = Erscheinenlassen des Inhalts des virtuellen Bildschirms durch verschieben des sichtbaren Bildes nach links **12** = Erscheinenlassen des Inhalts des virtuellen Bildschirms durch verschieben des sichtbaren Bildes nach unten **13** = Erscheinenlassen des Inhalts des virtuellen Bildschirms durch verschieben des sichtbaren Bildes nach oben

15,16,17,18 = Ebenso wie 10, 11, 12, 13, aber indem es über das sichtbare Bild hinweggleitet
20,21,22,23 = Wie die Modi 0, 1, 2, 3 , aber zusätzlich werden die Bilder ineinander verschlungen
30 =Horizontales ineinander verschlingen bei verschieben des sichtbaren Bildes
31 =Vertikales ineinander verschlingen bei verschieben des sichtbaren Bildes
40,41,42,43 = Erscheinenlassen des Inhalts des virtuellen Bildschirms entlang der Diagonalen

ANMERKUNG Der Parameter ***Etappen*** ermöglicht es, die Geschwindigkeit des Erscheinens des Bildes zu variieren. Je größer Etappen sind, desto langsamer erscheint das Bild. Der Parameter ***Etappen*** kann Werte zwischen 1 und 255 annehmen. Bei Verwendung eines Modus mit ineinander verschlingen, wird die Anzahl der Streifen durch die Differenz zwischen den Werten ***Breite_div*** und ***Höhe_div***, definiert in •**BHDiv**() bestimmt. Es wird empfohlen, eine Etappenzahl zu verwenden, welche jenachdem ob horizontal oder vertikal dargestellt wird, durch die Breite oder die Höhe dividierbar ist. Es wird empfohlen, eine Streifenzahl zu verwenden, welche jenachdem ob horizontal oder vertikal dargestellt wird, durch die Breite oder die Höhe dividierbar ist.

BEISPIEL

```
•VBschDef(200,200) •Bsch(1)
•Objekt(Foto)
•Bsch(0)
•WipeHor(1,200,150,300,300,0,0,20,0)
```

SIEHE AUCH •**VBschDef**(), •**Bsch**(), •**VBschLösch**(), •**Objekt**(), •**Bild**()

Alias: (Wmf) Befehl: Wmf

Darstellen eines Bildes (Vektorgrafik) vom Typ WMF

SYNTAX •**Wmf(*Name,Xfh,Yfh,Bf,Hf,Xbh,Ybh,Bb,Hb*)**

BESCHREIBUNG Erlaubt das Darstellen, Manipulieren und Ausschneiden von Vektorgraphik-Bildern von Typ WMF (Windows Meta File)

Name	:	Name des darzustellenden Bildes
Xfh,Yfh	:	Koordinaten der linken oberen Ecke der Fläche, in der das Bild dargestellt werden soll
Bf,Hf	:	Breite und Höhe der Fläche, in der das Bild dargestellt werden soll
Xbh,Ybh	:	Koordinaten der linken oberen Ecke der Fläche des Bildes, das dargestellt werden soll
Bb,Hb	:	Breite und Höhe der Fläche des Bildes, das dargestellt werden soll

ANMERKUNG Die Vektorgraphiken sind auf Wunsch in der Breite und Höhe veränderbar und an die vorgesehene Fläche anpassbar. Es ist also möglich, nur einen Teil des Bildes darzustellen und dieses zu skalieren. Um ein Bild als Ganzes und in seiner ursprünglichen Größe darzustellen, sind die Parameter des Rechteckes mit 0,0,1001,1001 festzulegen.

BEISPIEL

```
•Wmf(Objekt1.wmf,100,100,400,400,0,0,1001,1001)
•Wmf(Objekt1.wmf,200,200,500,300,400,400,100,100)
```

SIEHE AUCH •**Bild**(), •**Objekt**(), •**GrafikFormat**(), •**BildDim**(), •**BildEntf**(), •**BildMax**(), •**Transp**()

Alias: (Dico) Befehl: Wörterbuch

Aufrufen einer Wörterbuchfunktion

SYNTAX **•Wörterbuch(*Ausdruck,Buch,Pfad,Pos,Groß/Klein*)**

BESCHREIBUNG Sucht in dem Wörterbuch ***Buch*** die Definition für ***Ausdruck*** und kopiert sie in der Zeichenkette ***ewDef*** in das Verzeichnis ***Pfad***.

Ausdruck	:	Wort oder Ausdruck, der im Wörterbuch gesucht werden soll (max. 40 Buchstaben)
Buch	:	Wörterbuch, das durch formdico erstellt wurde
Pfad	:	Verzeichnis der Zeichenkettendatei
Pos	:	Name der Variablen, welche die Position des Wortes im Index beinhaltet
Groß/Klein	:	Wenn dieser Parameter nicht vorhanden ist (Grundeinstellung); dann wird das gesucht Wort in Großbuchstaben konvertiert, wenn das Zeichen **+** gesetzt ist, dann bleibt das Wort so erhalten, wie es ist.

ANMERKUNG Wenn man $$ als Suchbegriff eingibt, so schickt der Befehl **•Wörterbuch**() den Index des Wörterbuchs in die Zeichenkette ***ewTable***.

BEISPIEL

```
•Wörterbuch(OHM,Elektro,D:\tempo)
```

SIEHE AUCH Verwendung der Funktion ***formdico*** im Benutzerhandbuch

Alias: (Xfi) Befehl: Xfi

Vergleichen eines Antwortmodells mit einer Reihe von Zeichenketten

SYNTAX •**Xfi(*Antwortmod,AnfZKetteVar,EndZKetteVar,ErgebnisZKette*)**

BESCHREIBUNG Dieser Befehl ordnet den Zeichenkettenvariablen ***AnfZKetteVar*** und ***EndZKetteVar*** sukzessive die Antwortmodelle zu, die in Antwortmod enthalten sind. Sobald dieser Befehl ausgeführt ist, enthält die Variable ***ErgebnisZKette*** die Liste der Nummern der untersuchten Zeichenkettenvariablen, die vom Antwortmodell, das in Antwortmod enthalten ist, verstanden wurde. Die Liste ***ErgebnisZKette*** hat als Trennzeichen das Leerzeichen.

Antwortmod	:	Name der Zeichenkette, die das Antwortmodell enthält
AnfZKetteVar	:	Name der Zeichenkette einer Zeichenkettendatei, mit der der Befehl beginnt
EndZKetteVar	:	Name der Zeichenkette einer Zeichenkettendatei, die das Ende der Ausführung des Befehls bestimmt
ErgebnisZKette	:	Name der Zeichenkette, die das Ergebnis enthält

ANMERKUNG Das Antwortmodell muß die Form der "inversen Polnischen Schreibweise" haben.

BEISPIEL

```
[Kette:=´Autoren System EGO + .´]
•IPS(Kette,Modell)
•XFI(Modell, System{1},System{100},Resultat)
```

Alias: (Cligne) Befehl: ZeileNum

Übernimmt die Zeilennummer der Zeile in der dieser Befehl ausgeführt wird

SYNTAX •**ZeileNum(*VarName*)**

BESCHREIBUNG Die Variable ***VarName*** übernimmt bei Ausführung des Befehls die Zeilennummer.

VarName : Beinhaltet die Zeilennummer

SIEHE AUCH •**ModName**()

Alias: (Pmod) Befehl: ZeiMod

Festlegen des Zeichenmodus

SYNTAX •**ZeiMod(*Modus*)**

BESCHREIBUNG Definition des Zeichenmodus

Modus	***Zeichenmodus***
0	= transparent
1	= Zeichnen mit der Pinselfarbe (Standardeinstellung)
2	= Zeichen mit der inversen Pinselfarbe
3	= logisches UND zwischen Pinselfarbe und Hintergrund
4	= logisches UND zwischen inverser Pinselfarbe und Hintergrund
5	= logisches UND zwischen Pinselfarbe und inversem Hintergrund
6	= Inversion zwischen Pinselfarbe UND Hintergrund
7	= logisches XODER zwischen Pinselfarbe und Hintergrund
8	= Inversion zwischen Pinselfarbe XODER Hintergrund
9	= Inversion des Hintergrundes
10	= ogisches ODER zwischen Pinselfarbe und Hintergrund
11	= ODER zwischen inverser Pinselfarbe und Hintergrund
12	= ODER zwischen Pinselfarbe und inversem Hintergrund
13	= Inversion von Pinselfarbe ODER Hintergrund
14	= Zeichnet schwarz
15	= Zeichnet weiß

ANMERKUNG Der Modus für das Zeichnen paßt sich dem Pinsel und der Füllfarbe an.

BEISPIEL

```
•ZeiMod(7)
```

SIEHE AUCH •**PinselStil**()

Alias: (Ponct) Befehl: ZeiSetzung

Unterdrücken der Zeichensetzung

SYNTAX

- **`ZeiSetzung(+)`** oder
- **`ZeiSetzung(-)`**

BESCHREIBUNG Dieser Befehl ermöglicht die Definition einer Option zur Unterdrückung der Interpunktion in der Antwort des Schülers.

- ***ZeiSetzung***(+) : Aktiviert die Unterdrückung (Standardeinstellung)
- ***ZeiSetzung***(-) : Desaktiviert die Unterdrückung

SIEHE AUCH •`Sonderzeichen`(), •`KlBuchst`(), •`GrBuchst`()

Alias: (Zone) Befehl: Zone

Definieren einer rechteckigen Zone

SYNTAX •**Zone(*VarName,Xo,Yo,Breite,Höhe,Cursor*)**

BESCHREIBUNG Definieren einer rechteckigen Zone, deren obere linke Ecke die Koordinaten ***Xo,Yo*** hat und deren Breite gleich ***Breite*** und Höhe gleich ***Höhe*** ist.

VarName	:	Name einer globalen oder lokalen numerischen Variablen, die der Zone zugeordnet ist. Der der Variablen zugeordnete Wert gibt die Position und den Status der Maus in dieser Zone wieder. 0 = Die Maus befindet sich nicht in dieser Zone 1 = Die Maus befindet sich in dieser Zone 2 = Der Anwender hat (mit der Maus) in diese Zone geklickt 3 = Der Anwender hat (mit der Maus) in dieser Zone einen Doppelklick durchgeführt
Xo,Yo	:	Koordinaten der oberen linken Ecke der Zone
Breite	:	Breite der Zone
Höhe	:	Höhe der Zone
Cursor	:	Nummer des Cursors, der mit der Zone verknüpft ist

ANMERKUNG Sobald die Maus in die Zone eintritt, wird der assoziierten Variablen ***VarName*** der Wert 1 zugeordnet; wenn die Maus die Zone wieder verläßt, so wird der Wert der Variablen 0. Sobald die Zone angeklickt wird, verändert sich der Variablenwert auf 2. Bei einem Doppelklick nimmt die Variable den Wert 3 an.

Die assoziierte Variable kann global oder lokal sein. Wenn es sich um eine lokale Variable handelt, so ist die Zone lokal mit dem Modul verbunden, das sie erzeugt hat und verschwindet bei einer Verkettung mit einem anderen Modul. Man kann bis zu 600 Zonen gleichzeitig definieren.

Der Befehl •**TasteLösch**() ermöglicht das Löschen einer Zone.

BEISPIEL

```
•Zone(Menü,10,10,800,50)
```

SIEHE AUCH •**Taste**(), •**TasteAuswahl**() •**TasteCombo**(), •**TasteFont**(), •**TasteComboEdit**(), •**TasteGrafik**(), •**TasteListe**(), •**TasteLösch**() •**TasteMenü**(), •**TasteModus**(), •**TasteRadio**(), •**TasteScroll**()

MELDUNGEN UND FEHLERCODES

Sie finden hier eine Reihe von Meldungen und die damit korrespondierenden Nummern, welche bei der Ausführung eines Programms vom Schüler-Interpretor übermittelt werden, wenn ein Fehler auftreten sollte.
Die Systemvariable **ewErreur** beinhaltet die Fehlernummer und die Systemvariable **ewErreurInfo** beinhaltet den Parameter, der den Fehler bei der Ausführung ausgelöst hat (Name der nicht definierten Variablen, Name der nicht gefundenen Datei ...)

0 = Kein Fehler
1 = Überflüssige Zeichen
2 = Runde Klammern nicht vollständig
3 = Division durch 0
4 = Ungültige Operation
5 = Operator oder Zeichen nicht zulässig
6 = Funktion [ewErreurInfo] nicht gültig
7 = Anzahl der Parameter nicht richtig
8 = Zahl oder Variable wird erwartet
9 = Operator wird erwartet
10 = Es muß ein ganzzahliger Wert sein
11 = Anweisung nicht möglich, links von := muß eine Variable stehen
12 = Nicht kompatible Typen
13 = Fehler bei der Speicherunterstützung
14 = Zeichenkette nicht beendet
15 = Zeichenkette oder Zeichenketten-Variable wird erwartet
16 = Eckige Klammern nicht vollständig
17 = Lesefehler
18 = Schreibfehler
19 = Datei [ewErreurInfo] nicht gefunden
20 = Zeichenkettendatei [ewErreurInfo] bereits definiert
21 = Index-Fehler (Tabellen [ewErreurInfo] unterschiedlich festgelegt)
22 = Index außerhalb des Bereiches
23 = Tabelle [ewErreurInfo] nicht erstellt
24 = Erstellung von [ewErreurInfo] nicht gültig
25 = Öffnende Klammer fehlt
26 = Name einer Zeichenkettendatei oder einer Tabelle wird erwartet
27 = Bezugnahme auf eine nicht existierende Variable: [ewErreurInfo]
28 = Umbenennen von [ewErreurInfo] nicht möglich
29 = Löschen von [ewErreurInfo] nicht möglich
30 = Ungültiger Ausdruck

MELDUNGEN UND FEHLERCODES

(Fortsetzung)

31 = Das Argument der Funktion [ewErreurInfo] ist außerhalb des Bereiches
32 = Die Argumente der Funktion [ewErreurInfo] führen zu einem unbestimmten Ergebnis
33 = Die Funktion [ewErreurInfo] hat ein Überschreiten der Kapazität bewirkt
34 = Die Funktion [ewErreurInfo] hat ein Unterschreiten der Kapazität bewirkt
35 = Totaler Verlust der Genauigkeit für die Funktion [ewErreurInfo]
36 = Zeichenkette zu lang
37 = Es ist nicht möglich das Verzeichnis [ewErreurInfo] zu erstellen
38 = Es ist nicht möglich das Verzeichnis [ewErreurInfo] auszuwählen
39 = Es ist nicht möglich das Verzeichnis [ewErreurInfo] zu löschen
40 = Parameter in [ewErreurInfo] nicht in Ordnung oder unvollständig
41 = Unbekannter Fehler